

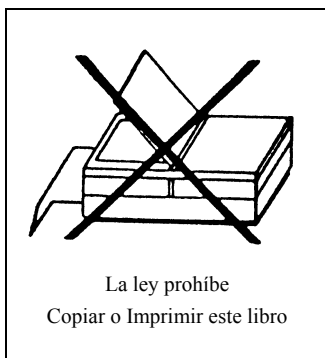
Implantación de Aplicaciones Web



Ra-Ma[®]

JUAN FERRER MARTÍNEZ

www.ra-ma.es/cf



IMPLANTACIÓN DE APLICACIONES WEB

© Juan Ferrer Martínez

© De la Edición Original en papel publicada por Editorial RA-MA

ISBN de Edición en Papel: 978-84-9964-160-7

Todos los derechos reservados © RA-MA, S.A. Editorial y Publicaciones, Madrid, España.

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es una marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeran o plagiaran, en todo o en parte, una obra literaria, artística o científica.

Editado por:

RA-MA, S.A. Editorial y Publicaciones
Calle Jarama, 33, Polígono Industrial IGARSA
28860 PARACUELLOS DE JARAMA, Madrid
Teléfono: 91 658 42 80
Fax: 91 662 81 39
Correo electrónico: editorial@ra-ma.com
Internet: www.ra-ma.es y www.ra-ma.com

Maquetación: Gustavo San Román Borrueco

Diseño Portada: Antonio García Tomé

ISBN: 978-84-9964-369-4

E-Book desarrollado en España en Septiembre de 2014

Implantación de Aplicaciones Web

JUAN FERRER MARTÍNEZ



Descarga de Material Adicional

Este E-book tiene disponible un material adicional que complementa el contenido del mismo.

Este material se encuentra disponible en nuestra página Web www.ra-ma.com.

Para descargarlo debe dirigirse a la ficha del libro de papel que se corresponde con el libro electrónico que Ud. ha adquirido. Para localizar la ficha del libro de papel puede utilizar el buscador de la Web.

Una vez en la ficha del libro encontrará un enlace con un texto similar a este:

“Descarga del material adicional del libro”

Pulsando sobre este enlace, el fichero comenzará a descargarse.

Una vez concluida la descarga dispondrá de un archivo comprimido. Debe utilizar un software descompresor adecuado para completar la operación. En el proceso de descompresión se le solicitará una contraseña, dicha contraseña coincide con los 13 dígitos del ISBN del libro de papel (incluidos los guiones).

Encontrará este dato en la misma ficha del libro donde descargó el material adicional.

Si tiene cualquier pregunta no dude en ponerse en contacto con nosotros en la siguiente dirección de correo: ebooks@ra-ma.com

*A mi esposa Ana,
su luz ilumina mi camino*

Índice

INTRODUCCIÓN	15
CAPÍTULO 1. CONCEPTOS GENERALES DE LA ARQUITECTURA DE APLICACIONES WEB	17
1.1 INTRODUCCIÓN	18
1.1.1 Concepto	18
1.1.2 ¿Por qué este concepto ha tomado tanta relevancia?	18
1.1.3 Problemas con las aplicaciones de escritorio.....	18
1.1.4 ¿Qué pasa con las aplicaciones de consola o modo texto?	19
1.1.5 La Web	19
1.1.6 Integración	20
1.2 APLICACIONES WEB VS. APLICACIONES DE ESCRITORIO	20
1.2.1 Ventajas del software web.....	20
1.3 ARQUITECTURA CLIENTE SERVIDOR. ELEMENTOS	21
1.4 ARQUITECTURA DE TRES NIVELES	23
1.5 PROTOCOLOS DE APLICACIÓN MÁS UTILIZADOS	24
1.5.1 El protocolo HTTP	25
1.5.2 El protocolo HTTPS.....	25
1.5.3 El protocolo FTP	25
1.5.4 El protocolo SMTP	26
RESUMEN DEL CAPÍTULO	26
EJERCICIOS PROPUESTOS	26
TEST DE CONOCIMIENTOS	27
CAPÍTULO 2. PREPARACIÓN DEL ENTORNO DE DESARROLLO Y SERVIDORES DE APLICACIONES WEB.....	29
2.1 ANÁLISIS DE REQUERIMIENTOS.....	30
2.2 SISTEMA OPERATIVO ANFITRIÓN: INSTALACIÓN Y CONFIGURACIÓN	31
2.2.1 Instalación de Windows XP	31
2.2.2 Instalación de Windows 7	37
2.2.3 Instalación de Ubuntu	40
2.3 INSTALACIÓN DEL SERVIDOR WEB APACHE, PREPROCESADOR PHP Y SGBD MYSQL EN WINDOWS	44
2.3.1 Configuración básica de Apache	47
2.3.2 Instalación y configuración del módulo PHP en Windows	47
2.3.3 Instalación de MySQL en Windows	52
2.4 INSTALACIÓN DEL SERVIDOR WEB APACHE, PREPROCESADOR PHP Y SGBD MYSQL EN UBUNTU.....	56
2.5 PHPMYADMIN.....	59
2.5.1 Utilizando PHPMyAdmin.....	60

2.6	XAMPP	63
2.6.1	Instalación de XAMPP en Windows	63
2.6.2	Instalación de XAMPP en Ubuntu	66
	RESUMEN DEL CAPÍTULO	68
	EJERCICIOS PROPUESTOS	68
	TEST DE CONOCIMIENTOS	69
	CAPÍTULO 3. GESTORES DE CONTENIDOS.....	71
3.1	DEFINICIÓN DE GESTOR DE CONTENIDOS O CMS	72
3.1.1	Historia de los CMS.....	72
3.1.2	Funcionamiento de un CMS	72
3.1.3	¿Para quién van destinados los CMS?.....	73
3.2	CONCEPTOS GENERALES DE UN CMS	73
3.3	ESTRUCTURA DE UN CMS	73
3.3.1	Rutina de trabajo.....	74
3.3.2	Gestión de usuarios	74
3.4	TIPOS DE CMS	75
3.5	LIMITACIONES DE LOS CMS	79
3.6	REQUERIMIENTOS PARA INSTALAR UN CMS	79
	RESUMEN DEL CAPÍTULO	79
	EJERCICIOS PROPUESTOS	80
	TEST DE CONOCIMIENTOS	80
	CAPÍTULO 4. BLOGGER	81
4.1	¿QUÉ ES UN BLOG?	82
4.2	ESTRUCTURA DE UN BLOG	82
4.2.1	La cabecera.....	82
4.2.2	El <i>post</i> o entrada.....	83
4.2.3	Los comentarios.....	83
4.2.4	El <i>blogroll</i>	83
4.2.5	Fecha/hora	84
4.2.6	Categorías/etiquetas	84
4.2.7	Enlace permanente o <i>permalink</i>	84
4.2.8	Sindicación	84
4.2.9	<i>Widgets</i> o <i>gadgets</i>	84
4.3	BLOGGER.....	85
4.3.1	Historia de Blogger	85
4.4	CREACIÓN DE UNA CUENTA DE CORREO ELECTRÓNICO CON GMAIL.....	86
4.6	CREACIÓN DE NUESTRO BLOG CON BLOGGER	87
4.6.1	Personalizar apariencia de nuestro <i>blog</i>	88
4.6.2	Creación de entradas en nuestro <i>blog</i>	89
4.6.3	Añadir <i>gadgets</i> a nuestro <i>blog</i>	94
4.6.4	Configuración avanzada de nuestro <i>blog</i>	96
4.6.5	Estadísticas de nuestro <i>blog</i>	97
4.7	CÓMO PONER MÚSICA EN NUESTRO BLOG: GOEAR.COM.....	97
4.8	CÓMO INSERTAR PRESENTACIONES A NUESTRO BLOG.....	100

RESUMEN DEL CAPÍTULO	101
EJERCICIOS PROPUESTOS	102
TEST DE CONOCIMIENTOS	102
CAPÍTULO 5. WORDPRESS.....	103
5.1 WORDPRESS.....	104
5.1.1 Historia de Wordpress.....	105
5.2 DAR DE ALTA NUESTRO BLOG.....	105
5.3 TEMAS	108
5.4 WIDGETS.....	109
5.5 FONDO	110
5.6 CREACIÓN DE ENTRADAS EN NUESTRO BLOG.....	112
5.7 ENCUESTAS	114
RESUMEN DEL CAPÍTULO.....	116
EJERCICIOS PROPUESTOS.....	116
TEST DE CONOCIMIENTOS	117
CAPÍTULO 6. WIKIS: MEDIAWIKI	119
6.1 DEFINICIÓN	120
6.2 COMPONENTES DE UNA WIKI	120
6.3 MEDIAWIKI.....	122
6.4 LICENCIAS DE USO DE MEDIAWIKI	122
6.5 REQUERIMIENTOS DE FUNCIONAMIENTO	123
6.6 INSTALACIÓN	123
6.6.1 Acceso a MediaWiki	124
6.6.2 Comprobación del entorno	125
6.6.3 Configuración de la base de datos.....	125
6.6.4 Configuración del sitio	127
6.6.5 Opciones de instalación.....	127
6.6.6 Extensiones	128
6.6.7 Carga de imágenes y archivos. Configuración avanzada	129
6.6.8 Completar la instalación.....	129
6.6.9 Problemas encontrados.....	130
6.7 USUARIOS Y GRUPOS.....	131
6.7.1 Sysops o administradores	132
6.7.2 Los burócratas.....	132
6.8 CREACIÓN DE CONTENIDOS	134
6.8.1 <i>Sidebar</i>	134
6.8.2 Pestañas	135
6.8.3 Enlaces del usuario	136
6.8.4 Crear una página nueva	138
6.9 PERSONALIZACIÓN DE LA INTERFAZ, PLANTILLAS.....	141
6.10 MECANISMOS DE SEGURIDAD INTEGRADOS.....	142
6.11 EDICIÓN.....	142
6.12 COPIAS DE SEGURIDAD	144

RESUMEN DEL CAPÍTULO.....	145
EJERCICIOS PROPUESTOS.....	146
TEST DE CONOCIMIENTOS	146
CAPÍTULO 7. MOODLE	147
7.1 DEFINICIÓN	148
7.2 LICENCIAS DE USO	148
7.3 REQUERIMIENTOS DE FUNCIONAMIENTO	149
7.4 INSTALACIÓN	149
7.5 CÓMO CREAR UN CURSO	159
7.6 ESTRUCTURA.....	160
7.6.1 Foros	161
7.6.2 Diarios	163
7.6.3 Apuntes, materiales o recursos.....	163
7.6.4 Tareas	163
7.6.5 Cuestionarios	164
7.6.6 Consultas.....	166
7.6.7 Encuestas	167
7.6.8 Chat	168
7.6.9 Glosario.....	169
7.6.10 Lección.....	169
7.6.11 Taller.....	169
7.7 CREACIÓN DE CONTENIDOS	170
7.8 PERSONALIZACIÓN DE LA INTERFAZ.....	170
7.8.1 Ajustes de temas	170
7.8.2 Selector de temas	172
7.9 USUARIOS Y GRUPOS.....	172
7.9.1 Administrador	172
7.9.2 Creador de cursos	172
7.9.3 Profesor.....	172
7.9.4 Estudiante	172
7.9.5 Invitado.....	173
7.10 CONTROL DE ACCESOS. RED MOODLE.....	173
7.11 INTEGRACIÓN DE MÓDULOS.....	174
7.11.1 Módulo de tareas	174
7.11.2 Módulo de consulta.....	174
7.11.3 Módulo foro.....	174
7.11.4 Módulo diario.....	175
7.11.5 Módulo cuestionario.....	175
7.11.6 Módulo recurso	175
7.11.7 Módulo encuesta.....	175
7.11.8 Módulo wiki	176
7.12 COPIAS DE SEGURIDAD	176
RESUMEN DEL CAPÍTULO.....	179
EJERCICIOS PROPUESTOS.....	179
TEST DE CONOCIMIENTOS	180

CAPÍTULO 8. FOROS: BBPRESS.....	181
8.1 DEFINICIÓN	182
8.2 LICENCIAS DE USO DE BBPRESS.....	182
8.3 REQUERIMIENTOS DE FUNCIONAMIENTO	182
8.4 INSTALACIÓN	183
8.5 ESTRUCTURA.....	186
8.6 PERSONALIZACIÓN DE LA INTERFAZ.....	188
8.7 PERFILES DE USUARIOS Y MECANISMOS DE SEGURIDAD INTEGRADOS.....	189
8.7.1 <i>Master key</i>	189
8.7.2 Administrador.....	189
8.7.3 Moderador	189
8.7.4 Miembro.....	189
8.7.5 Bloqueado	189
8.8 USUARIOS Y GRUPOS.....	190
8.9 INTEGRACIÓN DE MÓDULOS.....	192
8.9.1 Base de datos	192
8.9.2 <i>Cookies</i>	192
8.9.3 Registro.....	192
8.9.4 Funciones (opcional)	193
8.10 GESTIÓN DE TEMAS.....	193
8.11 PUBLICACIÓN.....	194
8.12 SINDICACIÓN DE CONTENIDOS.....	194
8.13 COPIAS DE SEGURIDAD	195
8.14 SELECCIÓN DE MODIFICACIONES A REALIZAR	196
8.15 INCORPORACIÓN Y ADAPTACIÓN DE FUNCIONALIDADES	197
RESUMEN DEL CAPÍTULO.....	197
EJERCICIOS PROPUESTOS	198
TEST DE CONOCIMIENTOS	198
CAPÍTULO 9. JOOMLA	199
9.1 DEFINICIÓN	200
9.1.1 Historia del proyecto	200
9.1.2 Versiones.....	200
9.2 LICENCIAS DE USO	201
9.3 REQUERIMIENTOS DE FUNCIONAMIENTO	202
9.4 DESCARGA DE JOOMLA	202
9.5 INSTALACIÓN DE JOOMLA.....	203
9.6 ESTRUCTURA DE JOOMLA	207
9.7 CATEGORÍAS	208
9.8 PLANTILLAS.....	211
9.9 USUARIOS Y GRUPOS.....	214
9.9.1 Niveles de acceso a la parte pública (<i>frontend</i>).	214
9.9.2 Niveles de acceso a la parte de administración (<i>backend</i>).	214
9.9.3 Crear o administrar usuarios en Joomla.....	214
9.10 MENÚS	216
9.11 MÓDULOS Y COMPONENTES	218

9.12	ARTÍCULOS	220
9.12.1	Lista de artículos	220
9.12.2	Editar artículos	221
9.12.3	Nuevo artículo	222
9.13	COPIAS DE SEGURIDAD	223
9.14	ACTUALIZACIONES	225
	RESUMEN DEL CAPÍTULO	225
	EJERCICIOS PROPUESTOS	226
	TEST DE CONOCIMIENTOS	226
	CAPÍTULO 10. IMPLANTACIÓN DE APLICACIONES DE OFIMÁTICA WEB	227
10.1	TIPOS DE APLICACIONES	228
10.1.1	Ventajas	228
10.1.2	Inconvenientes	229
10.1.3	Programas más populares	229
10.1.4	Google Docs.....	230
10.1.5	Zoho.....	233
10.1.6	Thinkfree	235
10.2	GOOGLE DOCS	237
10.2.1	Documentos de texto	237
10.2.2	Hojas de cálculo	237
10.2.3	Presentaciones	237
10.3	CÓMO DAR NOS DE ALTA EN EL SERVICIO GOOGLE DOCS	238
10.4	CREACIÓN DE DOCUMENTOS	239
10.4.1	La pantalla de inicio.....	239
10.4.2	Nuevo documento	239
10.4.3	Guardar un documento	240
10.5	COMPARTIR Y PUBLICAR	241
10.5.1	Compartir un elemento.....	242
10.5.2	Invitar a colaboradores y lectores.....	243
10.5.3	Gestionar el uso compartido	244
10.5.4	Público en la Web.....	245
10.5.5	Publicar un documento	246
10.6	DESCARGAR EL DOCUMENTO	248
	RESUMEN DEL CAPÍTULO	249
	EJERCICIOS PROPUESTOS	249
	TEST DE CONOCIMIENTOS	250
	CAPÍTULO 11. DISEÑO DEL CONTENIDO Y LA APARIENCIA DE DOCUMENTOS WEB.....	251
11.1	LENGUAJES DE MARCAS PARA REPRESENTAR EL CONTENIDO DE UN DOCUMENTO	252
11.1.1	¿HTML?	252
11.1.2	Herramientas para el desarrollo web.....	252
11.1.3	Nuestra primera página en HTML	253
11.1.4	Etiquetas HTML básicas	255
11.1.5	Etiquetas HTML avanzadas.....	261
11.1.6	Enlaces, vínculos o <i>links</i>	265

11.1.7	Imágenes	267
11.1.8	Tablas.....	270
11.1.9	Marcos (<i>frames</i>).....	276
11.1.10	Formularios	278
11.2	MODIFICACIÓN DE LA APARIENCIA DE UN DOCUMENTO WEB CON HOJAS DE ESTILOS	283
11.2.1	Usos de las css	283
11.2.2	Reglas de precedencia en los estilos	284
11.2.3	Sintaxis de las hojas de estilo en cascada.....	284
11.2.4	Atributos de las hojas de estilo	285
11.2.5	Definir estilos utilizando clases	293
11.2.6	Estilo en los enlaces	294
	RESUMEN DEL CAPÍTULO.....	296
	EJERCICIOS PROPUESTOS.....	296
	TEST DE CONOCIMIENTOS	297
	CAPÍTULO 12. PROGRAMACIÓN DE DOCUMENTOS WEB UTILIZANDO LENGUAJES DE SCRIPT DEL CLIENTE.....	299
12.1	DIFERENCIAS ENTRE LA EJECUCIÓN EN LADO DEL CLIENTE Y DEL SERVIDOR	300
12.2	MODELO DE OBJETOS DEL DOCUMENTO DOM.....	300
12.2.1	Introducción	300
12.2.2	Niveles del DOM y soporte.....	301
12.2.3	Navegadores compatibles con los estándares	301
12.3	JAVASCRIPT.....	302
12.3.1	Consideraciones iniciales.....	302
12.3.2	Primer ejemplo	302
12.4	VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT.....	303
12.4.1	Evento <i>onsubmit</i>	304
12.4.2	El objeto <i>this</i>	304
12.4.3	Validación de campos vacíos	305
12.4.4	Validación de campos numéricos	306
12.4.5	Validación de campos numéricos de longitud fija	307
12.4.6	Validación de campos de correo electrónico.....	307
12.5	COMPORTAMIENTOS DINÁMICOS. CAPTURA DE EVENTOS CON JAVASCRIPT	308
12.6	LIMITACIONES Y RIESGOS DE ATAQUES.....	310
	RESUMEN DEL CAPÍTULO.....	311
	EJERCICIOS PROPUESTOS.....	311
	TEST DE CONOCIMIENTOS	312
	CAPÍTULO 13. PROGRAMACIÓN DE DOCUMENTOS WEB UTILIZANDO LENGUAJES DE SCRIPT DE SERVIDOR: PHP	313
13.1	CLASIFICACIÓN	314
13.2	HERRAMIENTAS DE EDICIÓN DE CÓDIGO	314
13.2.1	Notepad ++	314
13.2.2	Dreamweaver	315
13.2.3	Eclipse.....	315
13.2.4	Aptana Studio	316

13.3	PHP	317
13.3.1	¿Qué se puede hacer con PHP?	317
13.3.2	Historia de PHP	317
13.4	INTEGRACIÓN CON LOS LENGUAJES DE MARCAS. SINTAXIS.....	318
13.5	COMENTARIOS	319
13.6	ELEMENTOS DEL LENGUAJE ESTRUCTURADO: TIPOS DE DATOS, VARIABLES, OPERADORES, ESTRUCTURAS DE CONTROL, SUBPROGRAMAS.....	320
13.6.1	Tipos de datos	320
13.6.2	Variables	321
13.6.3	Constantes.....	322
13.6.4	Expresiones	322
13.7	OPERADORES ARITMÉTICOS.....	323
13.8	OPERADORES DE ASIGNACIÓN.....	324
13.9	OPERADORES DE COMPARACIÓN	325
13.10	OPERADORES DE INCREMENTO/DECREMENTO	325
13.11	OPERADORES LÓGICOS	326
13.12	OPERADORES DE CADENAS.....	326
13.13	NÚMEROS ALEATORIOS	327
13.14	IF.....	328
13.15	ELSE.....	328
13.16	ELSEIF.....	329
13.17	WHILE.....	330
13.18	FOR.....	332
13.19	SWITCH	334
13.20	FUNCIONES INTEGRADAS Y DE USUARIO	336
13.20.1	Parámetros de las funciones.....	336
13.20.2	Pasar parámetros por referencia.....	336
13.20.3	Parámetros por defecto	337
13.20.4	Devolver valores.....	337
13.21	INCLUDE.....	339
13.22	REQUIRE.....	341
13.23	INCLUDE_ONCE() Y REQUIRE_ONCE()	341
13.24	ELEMENTOS DE ORIENTACIÓN A OBJETO	342
13.24.1	Las clases: <i>class</i>	342
13.24.2	Utilizar la clase	343
13.24.3	La variable <i>\$this</i>	343
13.25	GESTIÓN DE ERRORES.....	344
13.26	EXCEPCIONES EN PHP	344
13.27	FORMULARIOS. PROCESAMIENTO DE DATOS RECIBIDOS DESDE EL CLIENTE. MÉTODOS DE ENVÍO DE DATOS DESDE EL CLIENTE AL SERVIDOR	345
13.27.1	Envío a través del navegador.....	345
13.27.2	Método <i>get</i>	345
13.27.3	Método <i>post</i>	345
13.27.4	Identificación del método de envío	346
13.27.5	Diferencias ente los métodos <i>get</i> y <i>post</i>	346

13.27.6 La variable <code>\$_request</code>	346
13.28 USO DE LA FUNCIÓN <code>isset</code>	348
13.29 SESIONES	349
13.29.1 Trabajo con sesiones en PHP	349
13.30 PROBLEMAS DE AUTENTICACIÓN DE USUARIOS	353
13.31 MECANISMOS PARA MANTENER EL ESTADO ENTRE CONEXIONES	355
RESUMEN DEL CAPÍTULO	355
EJERCICIOS PROPUESTOS	356
TEST DE CONOCIMIENTOS	356
CAPÍTULO 14. ACCESO A BASES DE DATOS DESDE LENGUAJES DE SCRIPT DE SERVIDOR....	357
14.1 INTEGRACIÓN DE LOS LENGUAJES DE SCRIPT DE SERVIDOR CON LOS SISTEMAS GESTORES DE BASE DE DATOS	358
14.2 CONEXIÓN A BASES DE DATOS MYSQL CON PHP	358
14.3 CREACIÓN DE BASES DE DATOS EN MYSQL CON PHP	360
14.4 CREACIÓN DE TABLAS EN MYSQL CON PHP	361
14.5 CREACIÓN DE VISTAS EN MYSQL CON PHP. PROCEDIMIENTOS ALMACENADOS	363
14.6 INSERCIÓN DE INFORMACIÓN EN TABLAS MYSQL CON PHP	364
14.7 RECUPERACIÓN DE LA INFORMACIÓN DE LA BASE DE DATOS MYSQL DESDE UNA PÁGINA PHP	366
14.7.1 Consulta con varios resultados	366
14.7.2 Consulta con un solo resultado	367
14.8 MODIFICACIÓN DE LA INFORMACIÓN ALMACENADA. ACTUALIZACIONES Y BORRADOS	367
14.8.1 Actualizar campos de la tabla	367
14.8.2 Eliminar campos de una tabla	368
14.9 FUNCIONES ODBC	369
14.10 ADODB	372
14.10.1 Ejemplo MySQL	372
14.10.2 Aquí su equivalente en ADODB	372
14.10.3 Conectando a la base de datos	373
14.10.4 Ejecutando SQL	373
14.10.5 Obteniendo datos	373
14.10.6 Adoconnection	373
14.10.7 Otras funciones especiales	374
RESUMEN DEL CAPÍTULO	374
EJERCICIOS PROPUESTOS	375
TEST DE CONOCIMIENTOS	375
ÍNDICE ALFABÉTICO	379

Introducción

Este libro surge con el propósito de acercar al lector a los aspectos más importantes que encierra la **Implantación de Aplicaciones Web** ante la creciente demanda de personal cualificado para su administración. Con tal propósito, puede servir de apoyo también para estudiantes del Ciclo Formativo de Grado Superior de **Informática** y las **Ingenierías Técnicas**.

Hoy en día, existen muchos usuarios y profesionales de la Informática que discuten las ventajas e inconvenientes de algunas aplicaciones web y prefieren limitarse al uso exclusivo de una de ellas. Aquí no hay preferencia por ningún sistema en particular, ni se intenta compararlos para descubrir cuál es el mejor de todos, sino que se exponen sus principales características, manejo y métodos para conseguir la coexistencia entre ellos.

Para todo aquel que use este libro en el entorno de la enseñanza (Ciclos Formativos o Universidad) se ofrecen varias posibilidades: utilizar los conocimientos aquí expuestos para inculcar aspectos genéricos de las aplicaciones web o simplemente centrarse en preparar a fondo alguno de ellos. La extensión de los contenidos aquí incluidos hace imposible su desarrollo completo en la mayoría de los casos.

Ra-Ma pone a disposición de los profesores una guía didáctica para el desarrollo del tema que incluye las soluciones a los ejercicios expuestos en el texto. Puede solicitarlo a editorial@ra-ma.com, acreditándose como docente y siempre que el libro sea utilizado como texto base para impartir las clases.

1

Conceptos generales de la arquitectura de aplicaciones web

OBJETIVOS DEL CAPÍTULO

- ✓ Preparar el entorno de desarrollo y los servidores de aplicaciones web instalando e integrando las funcionalidades necesarias.
- ✓ Conocer las diferencias entre aplicaciones web y aplicaciones de escritorio.
- ✓ Identificar las diferencias entre los modelos de arquitectura cliente-servidor frente a modelos de arquitectura de tres capas.

1.1 INTRODUCCIÓN

1.1.1 CONCEPTO

Las aplicaciones web utilizan lo que se conoce como clientes livianos (*light clients*) los cuales no ejecutan demasiadas labores de procesamiento para la ejecución de la aplicación misma. Desde el punto de vista de la arquitectura se distinguen dos lados; uno es el **cliente**, donde se encuentra el usuario final utilizando la aplicación por medio de un navegador (como Internet Explorer o Mozilla Firefox). A través de este cliente web, el usuario interactúa con la aplicación localizada al otro lado, en el **servidor**, que es donde residen realmente los datos, reglas y lógica de la aplicación.

1.1.2 ¿POR QUÉ ESTE CONCEPTO HA TOMADO TANTA RELEVANCIA?

La esencia del concepto es no dejar que el cliente realice demasiadas tareas, sino solo lo necesario para que lleve a cabo su trabajo y dejar que en el lado del servidor se realicen las operaciones importantes: almacenamiento de datos, transacciones, reglas del negocio y la lógica del programa.

El concepto de aplicación web ha tomado una mayor relevancia con el auge de las redes locales y la popularidad de Internet, ofreciendo la oportunidad de acceso a dichas aplicaciones a través de computadores y otros dispositivos móviles. Internet ha elevado y extendido aún más el concepto de aplicación web para servir a usuarios ubicados en cualquier sitio donde se tenga acceso a Internet.

1.1.3 PROBLEMAS CON LAS APLICACIONES DE ESCRITORIO

Con la división del problema en dos partes, se logra centralizar la administración en general a un solo lado: el **servidor**. En él se resuelven una gran cantidad de problemas anteriormente encontrados en las aplicaciones de escritorio monousuario, como son:

- Duplicidad de datos por la falta de unificación de los mismos.
- Diseminación de la información y lógica en muchas partes (cada computador que la use).
- Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- La administración de la seguridad, ya que controlar el acceso de los usuarios a información no relevante o privada puede ser un caos.
- Dificultad para configurar cada una de las instalaciones (*deployments*) dependiendo de las necesidades de cada usuario.

1.1.4 ¿QUÉ PASA CON LAS APLICACIONES DE CONSOLA O MODO TEXTO?

Con aplicaciones de consola nos referimos a las construidas en plataformas tipo Cobol, RPG para AS400 y FoxPro, entre otras.

El concepto de las aplicaciones de consola es parecido al de una aplicación web con una arquitectura del tipo cliente-servidor en la cual el cliente también se puede considerar liviano. Aunque existen algunas diferencias como son:

- Protocolos de comunicación propios y no estándar, como ocurre en la Web con el protocolo HTTP y el concepto de URL.
- Formatos de intercambio propios y no estándar, como ocurre en la Web con el formato HTML o XML.

En el lado del cliente hay restricciones con las vistas, ya que es necesario instalar API específicas que no son estándar, portables o extensibles. En la Web solo se debe instalar un navegador para acceder a la aplicación.

La dependencia con el proveedor del software con respecto a la plataforma, arquitectura, hardware, sistema operativo y demás complementos que lleva consigo el “paquete” de la “solución” es inmensa en las aplicaciones de consola. En la Web la división por capas de las soluciones hace posible una independencia en todo sentido mucho mayor.

1.1.5 LA WEB

La Web se puede considerar como una plataforma o “sistema operativo” en el cual los recursos están distribuidos en la Red y están siendo extendidos en todo momento con posibilidades ilimitadas.

La Web se ha hecho popular con aplicaciones tales como clientes de correo, buscadores, portales, foros, *chats*, IRC, RSS, *blogs*, etc. Además de estas aplicaciones de propósito general, existe adicionalmente una gran diversidad de soluciones que se acomodan al ambiente web, como son: Administradores de contenido (CMS), Administrador de proyectos, Suites para trabajo colaborativo, Administración de relaciones con el cliente (CRM), ERP, etc.

La Web se reinventa día a día. Lo que ayer parecía imposible hoy es una realidad. Hace un año, o quizás meses, no se podía entender que hubiera tantas opciones para realizar una hoja de cálculo en una plataforma web como *docs.google.com*, o que pudiéramos jugar a un juego animado de construcción de mundos (tipo *Age of empires*) en *www.travian.net*, o que existiese un sistema de búsqueda mapas y direcciones como *maps.google.com*.

Google es uno de los que más ha ayudado al desarrollo y fomento de las aplicaciones web y sirve de infraestructura para llevar a cabo tales ideas.

Es cierto que la arquitectura cliente-servidor de la web ha ofrecido muchas ventajas, pero también es cierto que carece de la riqueza gráfica de las aplicaciones de escritorio que cuentan con controles inteligentes que dan mayor fluidez al trabajo del usuario. Esto ha sido resuelto con varias estrategias o tecnologías tales como AJAX, Flash y Web 2, entre otras. Así que en vez de ir perdiendo fuerza debido a la pobreza en sus interfaces gráficas, la Web busca alternativas que le permitan ofrecer todas sus ventajas, pero con la posibilidad de ofrecer controles visuales más amigables al trabajo del usuario.

1.1.6 INTEGRACIÓN

No se puede despreciar el enorme impacto que ha tenido el computador personal, o PC en la actualidad, el haber puesto al servicio de cualquier usuario el poder de una computadora en vez de una terminal simple ha potenciado una gran diversidad de usos.

Las aplicaciones de escritorio se han usado y se seguirán usando y tienen un campo enorme (sistemas CAD, CAM, suite de oficina, aplicaciones gráficas, juegos, utilidades o el mismo sistema operativo). No todo está en la Web, hay cosas que necesitan ejecutarse estrictamente en su máquina para aprovechar el poder que tiene a su alcance. Pero la fusión e integración de servicios de los computadores, las aplicaciones de escritorio y la extensión de las facultades de comunicación con las aplicaciones web que hacen posible Internet son la plataforma óptima que sirve de infraestructura para todos tipo de usuarios (empresarial, institucional o personal).

1.2 APLICACIONES WEB VS. APLICACIONES DE ESCRITORIO

1.2.1 VENTAJAS DEL SOFTWARE WEB

Hemos de indicar que todos y cada uno de los puntos que presentamos son plenamente “discutibles” y, por tanto, según la solución concreta sobre la que hablemos, es decir, el software concreto a utilizar, se podrán cumplir los siguientes puntos de forma total, parcial o nula.

- **No requiere instalar software especial (en los clientes).** En esencia, para acceder a un software web solo necesitamos disponer de un navegador de páginas web (Internet Explorer, Firefox, Opera, Chrome, etc.), los cuales suelen venir con el propio sistema operativo. No es necesario tener nada más. Debido a la arquitectura de las aplicaciones web, el navegador suele quedar relegado a mostrar la interfaz de usuario (menús, opciones, formularios, etc.), mientras que toda la compleja lógica de negocio se lleva en el lado del servidor.
- **Bajo coste en actualizar los equipos con una nueva versión.** Los navegadores web visualizan las páginas web que son servidas por el servidor web dinámicamente. En ese sentido, es el servidor quien ejecuta la mayor parte del código de la aplicación y suministra de forma centralizada las vistas (las páginas) a los navegadores conectados. En consecuencia, no hay que instalar nada en los puestos de trabajo, ya que la actualización se realiza en el servidor y automáticamente la ven todos los usuarios.
- **Acceso a la última y mejor versión.** Como consecuencia del punto anterior, se evita que pueda existir algún equipo que ejecute una versión diferente y desactualizada. Si existen ordenadores con distintas versiones del programa se pueden originar problemas de consistencia en la información o pérdida de funcionalidad.
- **Información centralizada.** En una aplicación web, no solamente la lógica de negocio está centralizada en el servidor, sino también los datos que se ubican en una base de datos centralizada (en ese servidor u otro destinado a tal fin). La centralización tiene la ventaja de facilitar el acceso a la misma.
- **Seguridad y copias de seguridad.** Este es un corolario del punto anterior, es decir, una consecuencia. Como disponemos de los datos centralizados es más fácil establecer y llevar el control de una política de copias de seguridad centralizada. Es más, al no ubicarse los datos en el puesto de trabajo, en caso de robo o incendio, la empresa no ha perdido información y puede desplegar rápidamente un nuevo puesto de trabajo (PC con un navegador web).

- **Movilidad.** Este es un concepto relativo y dependiente de la implantación concreta. Si el software está ubicado en un servidor web en Internet o bien disponemos de una intranet externalizada (extranet), cualquier usuario con un portátil y una conexión a Internet móvil podría acceder a la aplicación.
- **Reducción de costes en los puestos cliente (mayor longevidad).** Debido a que las páginas se ofrecen desde el servidor web (donde se suelen ejecutar la mayoría de los procesos y la lógica de negocio), el equipo cliente queda relegado a mostrar los resultados y formularios, para lo cual no es necesario un hardware potente en los puestos de trabajo, lo que se traduce en reducción de costes y una mayor longevidad en el uso de los mismos (no hay que cambiar el hardware de los puestos porque ahora se requieran operaciones más complejas).

Sin embargo no todo son ventajas. Debemos recordar que en el mundo real de los requisitos y restricciones no existe la solución perfecta, sino la más o menos adecuada al caso planteado. Por ello, una solución web también tiene sus inconvenientes, unos derivados del modelo web y otros como consecuencia de cómo se implante.

1.3 ARQUITECTURA CLIENTE SERVIDOR. ELEMENTOS

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con los que se cuentan y que se ponen a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor; de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de solo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

Características

En la arquitectura C/S el remitente de una solicitud es conocido como cliente. Sus características son:

- ✓ Es el que inicia solicitudes o peticiones. Tiene, por tanto, un papel activo en la comunicación (dispositivo maestro o amo).
- ✓ Espera y recibe las respuestas del servidor.
- ✓ Por lo general, puede conectarse a varios servidores a la vez.
- ✓ Normalmente, interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.
- ✓ Al contratar un servicio de red, se debe tener en cuenta la velocidad de conexión que se le otorga al cliente y el tipo de cable que utiliza.

Al receptor de la solicitud enviada por el cliente se conoce como servidor. Sus características son:

- ✓ Al iniciarse espera a que le lleguen las solicitudes de los clientes. Desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).
- ✓ Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- ✓ Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- ✓ No es frecuente que interactúen directamente con los usuarios finales.

Ventajas

- ✓ **Centralización del control:** los accesos, recursos y la integridad de los datos son controlados por el servidor, de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- ✓ **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- ✓ **Fácil mantenimiento:** al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- ✓ **Tecnologías:** existen algunas suficientemente desarrolladas, diseñadas para el paradigma de C/S, que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz y la facilidad de empleo.

Desventajas

- ✓ La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P, como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda que se tiene.

- ✓ El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está caído las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.
- ✓ El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente, se necesita software y hardware específicos, sobre todo en el lado del servidor para satisfacer el trabajo. Por supuesto, esto aumentará el coste.
- ✓ El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

Ejemplos. La mayoría de los servicios de Internet son tipo de cliente-servidor. La acción de visitar un sitio web requiere una arquitectura cliente-servidor, ya que el servidor web sirve las páginas web al navegador (al cliente). Al leer este artículo en Wikipedia, la computadora y el navegador web del usuario serían considerados un cliente; y las computadoras, las bases de datos y los usos que componen Wikipedia serían considerados el servidor. Cuando el navegador web del usuario solicita un artículo particular de Wikipedia, el servidor de Wikipedia recopila toda la información a mostrar en su base de datos, la articula en una página web y la envía de nuevo al navegador web del cliente.

Otro ejemplo podría ser el funcionamiento de un juego *on line*. Si existen dos servidores de juego, cuando un usuario lo descarga y lo instala en su computadora pasa a ser un cliente. Si tres personas juegan en un solo computador existirían dos servidores, un cliente y tres usuarios. Si cada usuario instala el juego en su propio ordenador existirían dos servidores, tres clientes y tres usuarios.

1.4 ARQUITECTURA DE TRES NIVELES

En la arquitectura en tres niveles existe un nivel intermedio. Esto significa que la arquitectura generalmente está compartida por:

- Un **cliente**, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
- El **servidor de aplicaciones** (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El **servidor de datos**, que proporciona al servidor de aplicaciones los datos que éste le solicitó.

El uso masivo del término arquitectura en tres niveles también denota las siguientes arquitecturas:

- Aplicación compartida entre un cliente, un software intermedio y un servidor empresarial.
- Aplicación compartida entre un cliente, un servidor de aplicaciones y un servidor de base de datos empresarial.

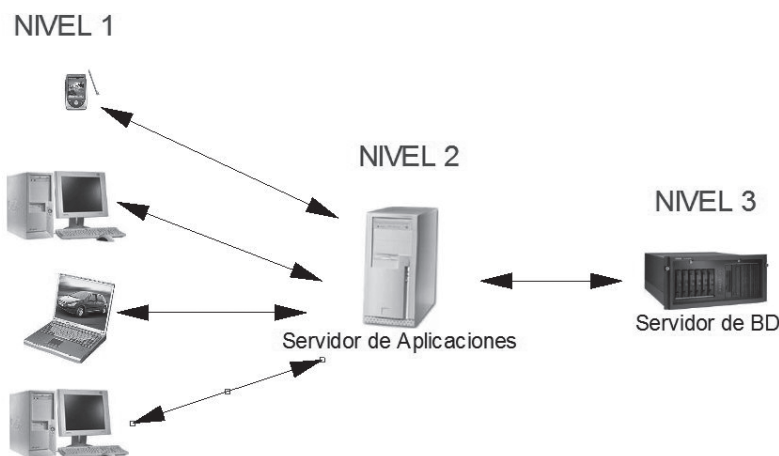


Figura 1.1. Arquitectura de 3 niveles

La arquitectura en dos niveles es, por tanto, una arquitectura cliente-servidor en la que el servidor es polivalente, es decir, puede responder directamente a todas las solicitudes de recursos del cliente.

Sin embargo, en la arquitectura en tres niveles las aplicaciones al nivel del servidor son descentralizadas de uno a otro, es decir, cada servidor se especializa en una determinada tarea, (por ejemplo, servidor web/servidor de bases de datos). La arquitectura en tres niveles permite:

- Un mayor grado de flexibilidad.
- Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel.
- Mejor rendimiento, ya que las tareas se comparten entre servidores.



Hay que tener en cuenta que los distintos niveles representados como equipos físicos distintos podrían ser llevados a cabo por programas “servidores” dentro de un mismo servidor físico. Es decir, un software que funcionase como servidor de aplicaciones y otro software como servidor de bases de datos.

1.5 PROTOCOLOS DE APLICACIÓN MÁS UTILIZADOS

Un **protocolo** es un método estándar que permite la comunicación entre procesos (que potencialmente se ejecutan en diferentes equipos), es decir, es un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red. Existen diversos protocolos de acuerdo a cómo se espera que sea la comunicación. Algunos protocolos, por ejemplo, se especializarán en el intercambio de archivos, como el FTP (*File Transfer Protocol*, Protocolo de transferencia de ficheros); otros pueden utilizarse simplemente para administrar el estado de la transmisión y los errores (como es el caso de ICMP), etc.

En Internet, los protocolos utilizados pertenecen a una sucesión de protocolos o a un conjunto de protocolos relacionados entre sí. Este conjunto de protocolos se denomina TCP/IP. Entre otros, contiene los siguientes protocolos:

1.5.1 EL PROTOCOLO HTTP

Desde 1990, el protocolo HTTP (*Hiper Text Transfer Protocol*, Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet. La versión 0.9 solo tenía la finalidad de transferir los datos a través de Internet (en particular páginas web escritas en HTML). La versión 1.0 del protocolo (la más utilizada) permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME.

El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL (*Uniform Resource Locator*, localizador uniforme de recursos).

La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

- El navegador realiza una solicitud HTTP.
- El servidor procesa la solicitud y después envía una respuesta HTTP.

1.5.2 EL PROTOCOLO HTTPS

El protocolo seguro de Transferencia de hipertexto (HTTPS, *Hiper Text Transfer Protocol Secure*) es la versión segura del protocolo HTTP. La diferencia es que HTTPS permite realizar transacciones de forma segura. Por lo tanto, podremos desarrollar actividades de tipo *e-commerce*, acceso a cuentas bancarias *on line*, tramites con la administración pública, etc.

En los navegadores comunes como Firefox, Explorer o Chrome, cuando estamos empleando un protocolo HTTPS podemos ver el icono de un candado que aparece en la barra principal de nuestro navegador. Además, en la barra de direcciones podremos ver que “http://” será sustituido por “https://”.

Y, ¿cómo funciona la conexión exactamente? ¿Por qué es más segura? Básicamente, lo que ocurre es que la página web codifica la sesión con certificado digital. De este modo, el usuario tiene ciertas garantías de que la información que envíe desde dicha página no podrá ser interceptada y utilizada por terceros.

Estos certificados de seguridad son conocidos como SSL. Cuando estos están instalados en la página web veremos el candado del que hablábamos anteriormente. Por otro lado, si hay instalados Certificados de Validación Extendida, además del candado, los usuarios podremos ver que la barra de URL del navegador toma un fondo verdoso.

1.5.3 EL PROTOCOLO FTP

El protocolo FTP (*File Transfer Protocol*, Protocolo de transferencia de archivos) es, como su propio nombre indica, un protocolo para transferir archivos.

La implementación del FTP se remonta a 1971, cuando se desarrolló un sistema de transferencia de archivos (descrito en RFC141) entre equipos del Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology). Desde entonces, diversos documentos de RFC han mejorado el protocolo básico, pero las innovaciones más importantes se llevaron a cabo en julio de 1973.

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP. El objetivo del protocolo FTP es:

- Permitir que equipos remotos puedan compartir archivos.
- Permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor.
- Permitir una transferencia de datos eficaz.

1.5.4 EL PROTOCOLO SMTP

El protocolo SMTP (*Simple Mail Transfer Protocol*, Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII. El tamaño máximo permitido para estas líneas es de 1.000 caracteres.



RESUMEN DEL CAPÍTULO

En este primer capítulo hemos visto una introducción a las aplicaciones web, hemos visto las principales ventajas y diferencias con respecto al resto de aplicaciones, las llamadas aplicaciones de escritorio y las aplicaciones de consola.

También se han planteado diferentes modelos de arquitectura para implementar las aplicaciones comentadas: el modelo cliente-servidor y el modelo de tres capas.



EJERCICIOS PROPUESTOS

1. ¿Qué ventajas se ven con respecto al uso de las aplicaciones web?
2. ¿Por qué cree que en la actualidad tienen tanta importancia las aplicaciones web?
3. ¿Qué problemas encuentra en las aplicaciones de escritorio que se solucionen mediante el uso de aplicaciones web?
4. ¿Qué ventajas tienen las aplicaciones web con respecto a las aplicaciones de consola?
5. Enumere las ventajas de las aplicaciones web frente a las aplicaciones de escritorio.
6. Defina la arquitectura cliente-servidor.
7. Características principales de la arquitectura cliente-servidor.
8. Problemas de la arquitectura cliente-servidor.
9. Defina la arquitectura de tres niveles.
10. Características principales de la arquitectura de tres niveles.
11. Problemas de la arquitectura de 3 niveles.



TEST DE CONOCIMIENTOS



1 La arquitectura, que es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores y los demandantes, se llama:

- a) Arquitectura secuencial.
- b) Arquitectura de tres niveles.
- c) Arquitectura cliente-servidor.
- d) No existe ese tipo de modelo.

2 Con las aplicaciones de escritorio:

- a) Son independientes del sistema operativo que tengamos.
- b) Tenemos la información centralizada.
- c) Es más fácil implementar modificaciones en el programa.
- d) Es más sencillo el respaldo de información.
- e) Todas son ciertas.
- f) Todas son falsas.

3 Con las aplicaciones web:

- a) Son independientes del sistema operativo que tengamos.
- b) Tenemos la información centralizada.
- c) Es más fácil implementar modificaciones en el programa.
- d) Es más sencillo el respaldo de información.
- e) Todas son ciertas.
- f) Todas son falsas.

4 Todas las aplicaciones que hay en el mercado son web:

- a) Cierto, ya que como se ha visto en el capítulo, es el mejor sistema.
- b) Cierto, ya que el resto no se distribuyen en la actualidad.
- c) Falso, ya que hay ciertas aplicaciones web que funcionan sin conexión a Internet.
- d) Falso, ya que hay otro tipo de aplicaciones que no son web.

5 El modelo de tres capas NO puede implementarse en un solo equipo:

- a) Cierto, ya que necesitas obligatoriamente tres equipos: un cliente, un servidor de aplicación y un servidor de bases de datos.
- b) Cierto, ya que necesitas tener conexión a Internet de forma obligatoria en todos los equipos.
- c) Falso, ya que lo que aquí se llaman equipos, pueden ser realizados por programas: un programa cliente, un programa servidor de aplicación y un programa servidor de bases de datos.
- d) Falso, ya que si es un equipo muy caro sí podrá implementarse en una sola máquina.

6 El modelo de cliente-servidor NO puede implementarse en un solo equipo:

- a) Cierto, ya que necesitas obligatoriamente dos equipos: un cliente y un servidor.
- b) Cierto, ya que necesitas tener conexión a Internet de forma obligatoria en todos los equipos.
- c) Falso, ya que las funciones destinadas a los equipos pueden ser realizadas por programas: un programa cliente y un programa servidor.
- d) Falso, ya que si es un equipo muy caro sí podrá implementarse en una sola máquina.

2

Preparación del entorno de desarrollo y servidores de aplicaciones web

OBJETIVOS DEL CAPÍTULO

- ✓ Instalar sistemas operativos en máquinas en las que irán los servidores web.
- ✓ Instalar Servidores Apache, PHP y MySQL en las máquinas.
- ✓ Instalar PHPMyAdmin.
- ✓ Instalar XAMPP y LAMPP.

2.1 ANÁLISIS DE REQUERIMIENTOS

Para poder usar los servidores web que vamos a utilizar de aquí en adelante vamos a necesitar una máquina con un sistema operativo Windows o Linux e instalar dichos servidores web y de base de datos en nuestra máquina.

Con respecto a los requerimientos hardware, cualquier equipo existente en la actualidad cumple de sobra los requisitos mínimos para convertirse en un servidor web para un nivel de tráfico “normal” y unas cuantas páginas web. No estaría de más utilizar cualquiera (o todos) de los sistemas de respaldo existentes: utilizar discos duros en espejo, copias de seguridad de la información de forma periódica, etc.

Ahora bien, si prevemos un tráfico intenso y sitios con PHP, Bases de datos, foros, etc., quizás necesitemos un procesador bastante rápido, con mucha memoria RAM y discos duros sólidos. En algunos casos se podría separar el servidor web del servidor de las bases de datos en dos (o más) servidores distintos.

El sistema operativo que instalemos en nuestro/s servidor/es dependerá también del uso que se le vaya a dar y del presupuesto con el que contemos. Hay que tener en cuenta que la mayoría de los servidores web que hay cuentan con un sistema operativo Unix/Linux, principalmente por su robustez, estabilidad y seguridad frente ataques externos e internos. Si se es pro Windows también se puede elegir cualquiera de las versiones del sistema operativo de Microsoft (preferiblemente que sean Server).

El sistema de ficheros utilizado irá en función al sistema operativo que elijamos para nuestro servidor. Si nos decantamos por Windows deberá ser NTFS (no utilizar FAT ni FAT32). En el caso de Unix/Linux el abanico es más amplio: podemos utilizar ext (recomendado ext4), ReiserFS, JFS, etc.

Con respecto al Sistema Gestor de Bases de Datos ocurre lo mismo que con el hardware necesario, variará en función de las necesidades de nuestro servidor, del presupuesto y de nuestro conocimiento sobre dichos gestores. Existen gestores gratuitos como PostgreSQL o MySQL y gestores de pago como Oracle o SQLServer.

¿Qué decir de la conexión a Internet de nuestro servidor? Quizás esté aquí el mayor cuello de botella de todo el proceso. Debemos asegurar una conexión que esté libre de fallos y sea lo más rápida posible. Habrá que dar mayor prioridad a la velocidad de subida de la conexión frente a la de bajada. Es muy importante la velocidad de subida, ya que la velocidad con la que los clientes pueden acceder e interactuar con nuestro servidor estará principalmente ligada a la velocidad de subida que pueda proporcionar dicho servidor (sobre todo frente a varios accesos concurrentes al servidor web).

Con respecto al cliente, aquí sí que es verdad que nos valdrá cualquier dispositivo, sea un ordenador, un *tablet*, un *smartphone*, una televisión, una consola, etc. Tan solo van a necesitar una conexión a Internet y un programa que permita comunicarse con el servidor, como un navegador web, un gestor de correo electrónico, etc.

En este capítulo vamos a ver una introducción rápida de cómo instalar Windows XP, Windows 7 y Ubuntu para utilizarlos como servidores web (son los sistemas operativos más comunes que vamos a encontrar en el aula). A su vez también vamos a ver cómo instalar y configurar un servidor web Apache, PHP y MySQL para dichos sistemas operativos de nuestro servidor. Sobra comentar que podemos utilizar también cualquiera de estos sistemas operativos para los clientes.

2.2 SISTEMA OPERATIVO ANFITRIÓN: INSTALACIÓN Y CONFIGURACIÓN

2.2.1 INSTALACIÓN DE WINDOWS XP

Windows XP ha sido el sistema operativo de Microsoft más utilizado, por lo menos hasta la aparición de Windows 7. Ni siquiera la aparición de Windows Vista hizo perder la hegemonía de Windows XP.

Los requisitos mínimos para poder instalar Windows XP en nuestra máquina son: procesador Pentium a 233 MHz (se recomienda 600 Mhz), 64 MB de RAM (se recomienda 128 MB, 1,5 GB de espacio disponible en el disco duro y una unidad de CD-ROM o DVD-ROM

Necesitaremos introducir el CD-ROM en la unidad de CD/DVD y reiniciar el ordenador para comenzar la instalación de Windows XP. Con el tipo de instalación que se presenta a continuación se borrarán todos los archivos de nuestro ordenador, así que, en el caso de que se trate de una reinstalación, debemos guardar todos los datos que tengamos en él en soportes externos como CD, DVD o dispositivos de memoria USB, antes de comenzar la instalación.

Iniciamos la instalación con el mensaje “Pulse cualquier tecla para iniciar desde el CD”, pulsamos cualquier tecla y comienza la primera parte de la instalación, esta parte se realiza en modo texto. De momento no es necesaria la intervención de usuario.

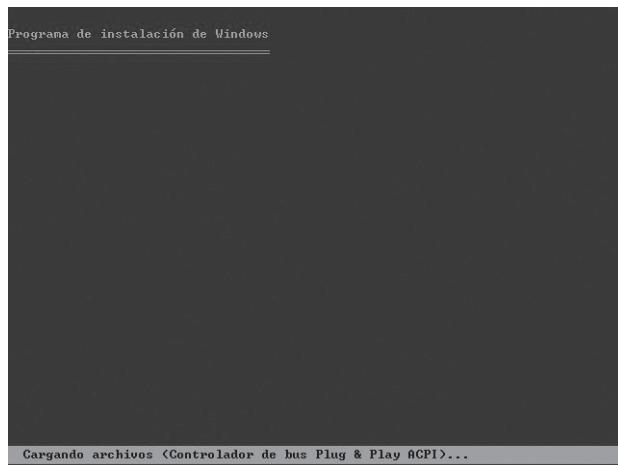


Figura 2.1



Figura 2.2

Primera decisión a tomar, podemos instalar o recuperar una instalación anterior utilizando la consola de recuperación. En nuestro caso vamos a seleccionar instalación, por tanto debemos pulsar la tecla **Intro** (Figura 2.3). Acto seguido nos aparecerá el contrato de licencia de Windows XP, una vez lo hayamos leído pulsamos **F8** para aceptar dicho contrato y proseguir con la instalación (Figura 2.4).

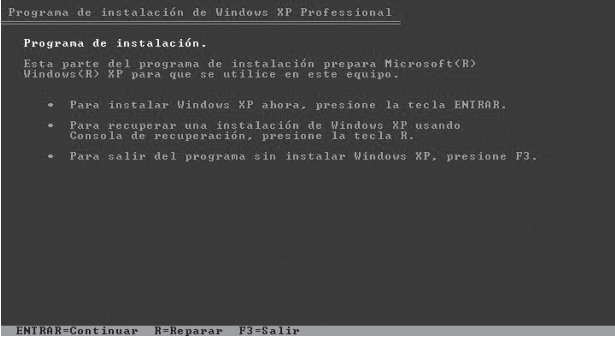


Figura 2.3

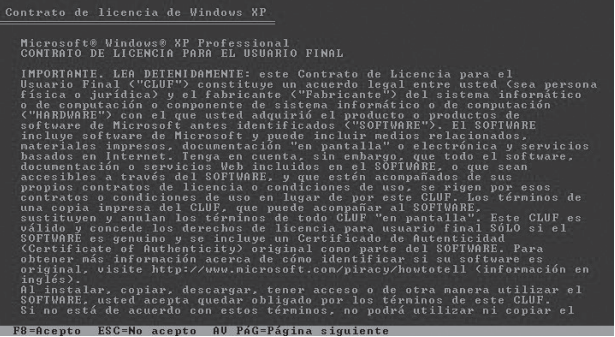


Figura 2.4

Nos encontramos ahora con una pantalla donde nos pide que seleccionemos la unidad de disco o partición donde vamos a instalar Windows. Hay que tener especial cuidado si tenemos más de una unidad física o lógica, pues si elegimos mal la unidad podríamos perder datos importantes (Figura 2.5). Una vez elegida la unidad donde instalaremos Windows XP debemos seleccionar el sistema de archivos que tendrá la unidad. Se recomienda utilizar el sistema de archivos NTFS y el tipo de formato rápido, si tenemos dudas sobre integridad del disco en el que vamos a instalar debemos seleccionar el tipo de formato normal (siempre con el sistema NTFS), ya que este formato comprueba si existen sectores defectuosos en dicho disco (Figura 2.6).

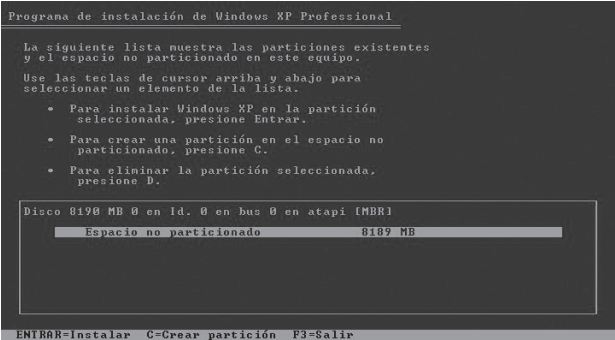


Figura 2.5

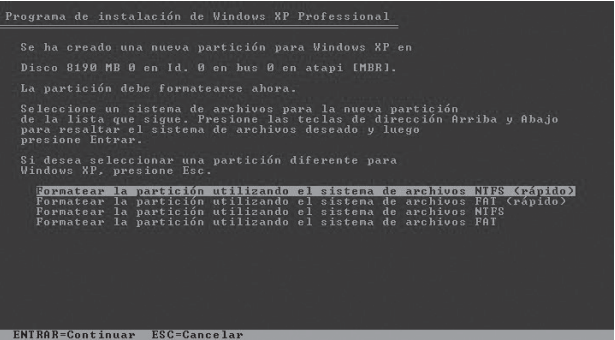


Figura 2.6

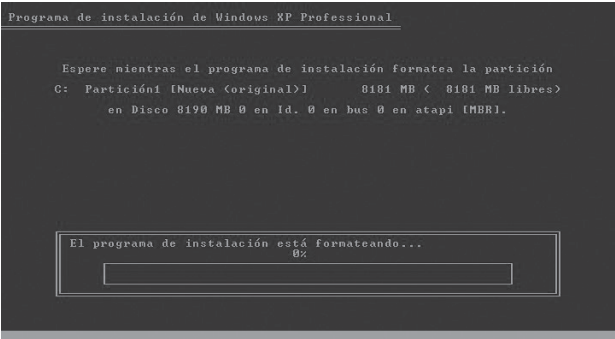


Figura 2.7

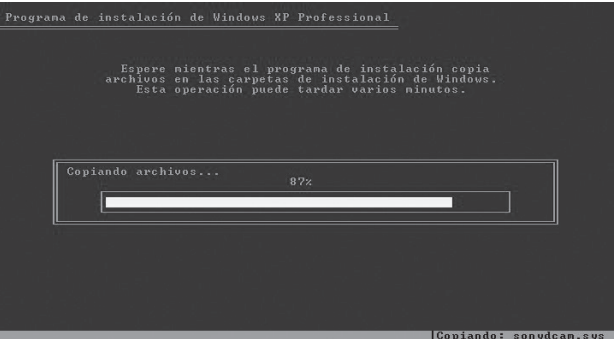


Figura 2.8

Una vez formateada la unidad, se copian una serie de ficheros que nos servirán para la segunda parte de la instalación. Para que se pueda iniciar esta parte debe reiniciarse el equipo (Figura 2.9).

En este caso cuando aparezca el mensaje “Pulse cualquier tecla...” no debemos pulsar ninguna tecla, ya que en ese caso lanzaríamos la instalación otra vez desde el principio. Comentar que esta segunda parte de la instalación se realizara en modo grafico (Figura 2.10).

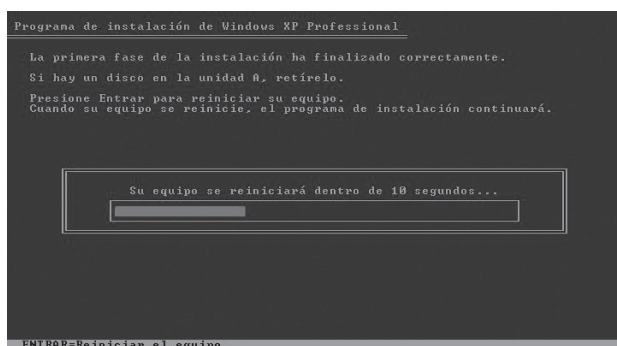


Figura 2.9



Figura 2.10

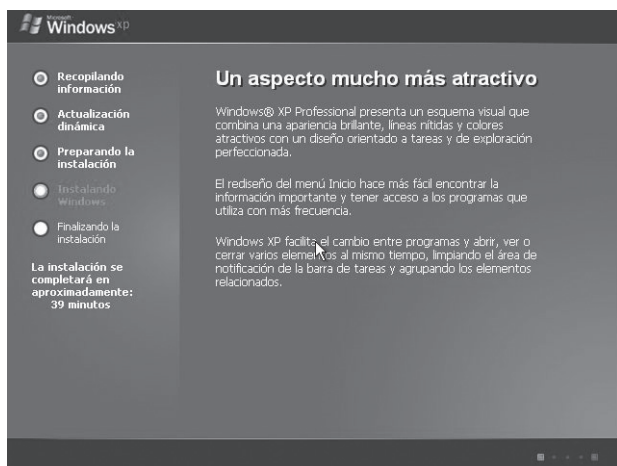


Figura 2.11

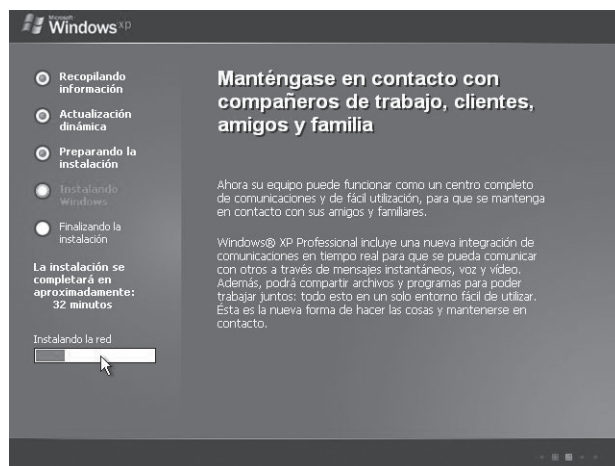


Figura 2.12

Tras unos minutos sin pedir la intervención del usuario, y después de habernos mostrado las “excelencias” de Windows XP, tendremos que configurar las opciones de región e idioma. En nuestro caso tanto la configuración regional y de idioma son correctos, por tanto, no deberemos hacer ningún cambio (Figura 2.13).

Acto seguido nos pide el nombre y la organización. Rellenamos estos datos y continuamos (Figura 2.14).

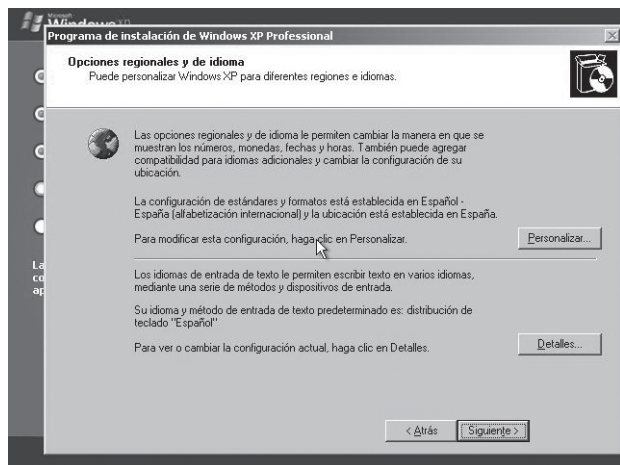


Figura 2.13

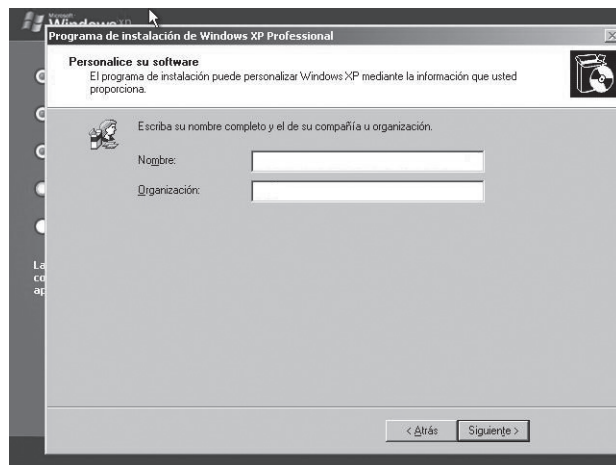


Figura 2.14

En este punto debemos introducir la clave del producto de nuestra copia de Windows XP, esta clave suele venir en una pegatina en el lateral del ordenador, o la caja de nuestro sistema operativo (Figura 2.15).

Después de introducir la clave, debemos poner un nombre a nuestro equipo, este nombre será mediante el cual se conocerá a este equipo en nuestra red local. También debemos proporcionar una contraseña para el usuario administrador, si no queremos contraseña podemos dejar estos campos en blanco (Figura 2.16).

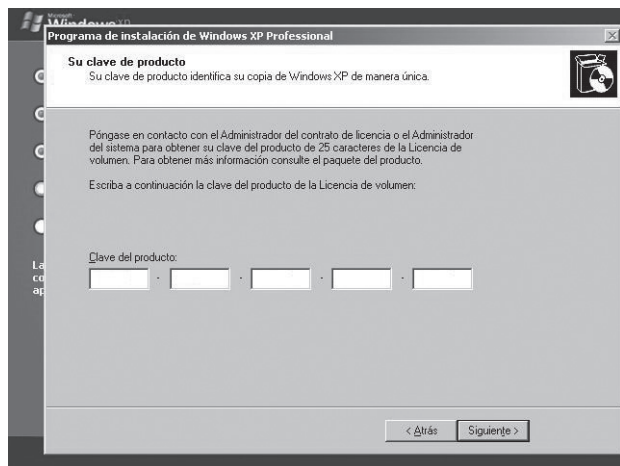


Figura 2.15

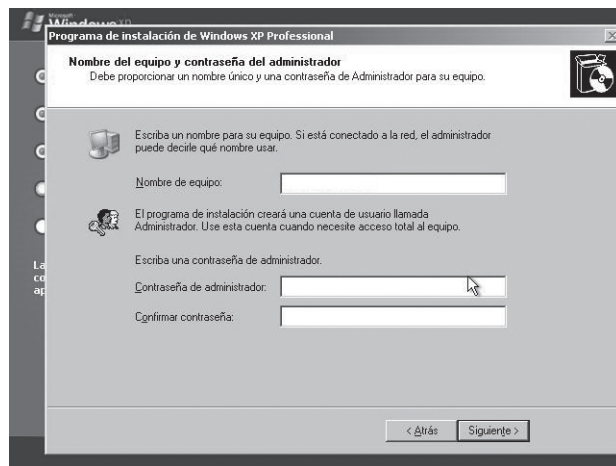


Figura 2.16

En la siguiente pantalla podemos establecer la fecha y la hora, así como la zona horaria en la que nos encontramos. También podemos activar que cambie la hora automáticamente según el horario de verano e invierno (Figura 2.17).

Después de esto nos encontramos con la configuración de la red, en principio, podemos dejarla en la configuración típica, ya que instalará todos los protocolos y servicios necesarios en nuestra máquina, tanto para utilizarla en una red local, como para el uso de Internet (Figura 2.18).

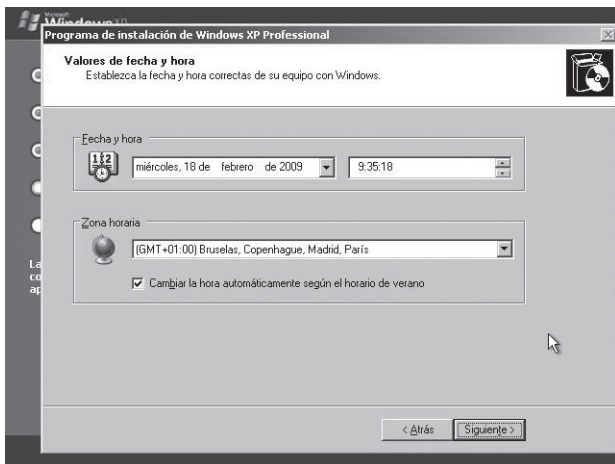


Figura 2.17

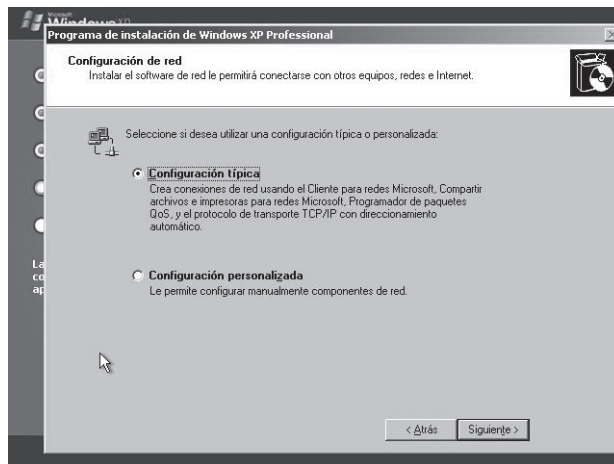


Figura 2.18

Estamos llegando casi al final de la instalación, por ultimo debemos introducir el grupo de trabajo o el dominio al que pertenecemos. Si el ordenador no forma parte de un grupo, o no sabemos al grupo al que pertenecemos, podemos dejar el que viene por defecto y continuar con la instalación. Una vez instalado podremos cambiar el grupo o el dominio al que pertenecemos de forma sencilla (Figura 2.19).

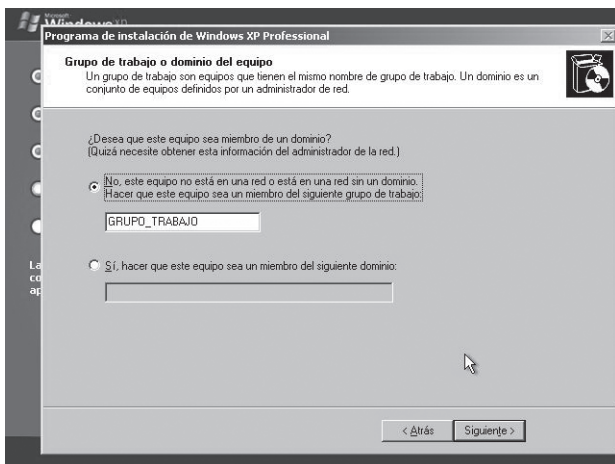


Figura 2.19

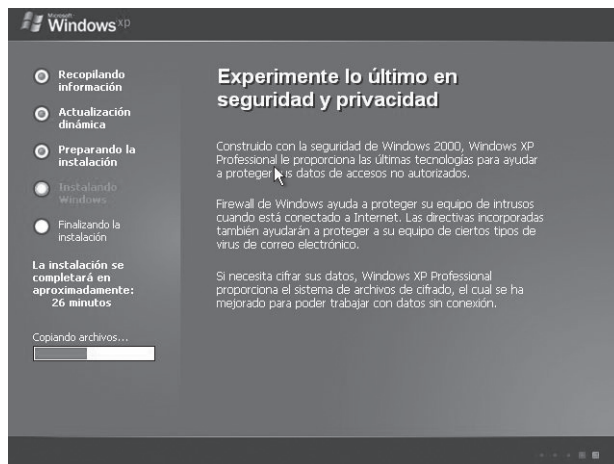


Figura 2.20

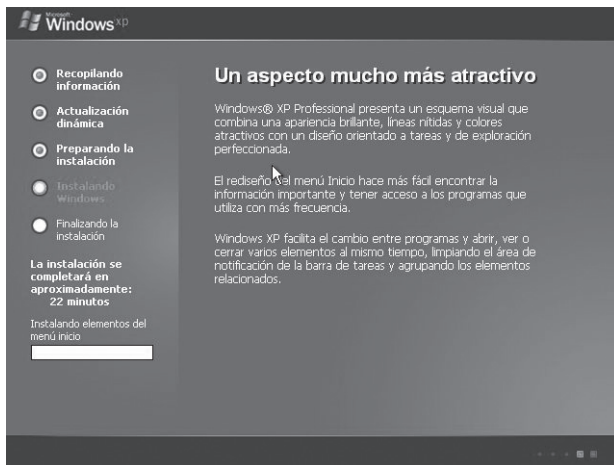


Figura 2.21



Figura 2.22

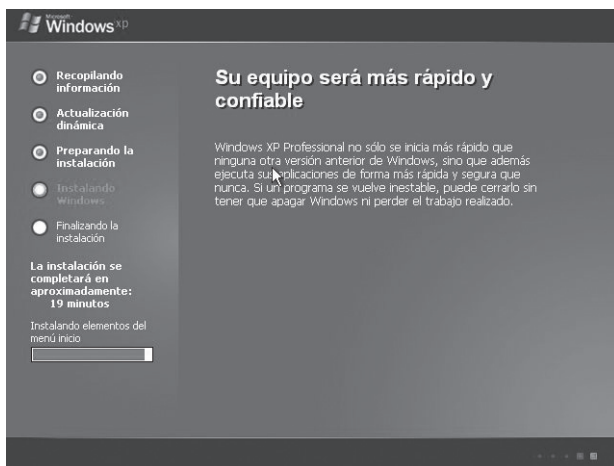


Figura 2.23

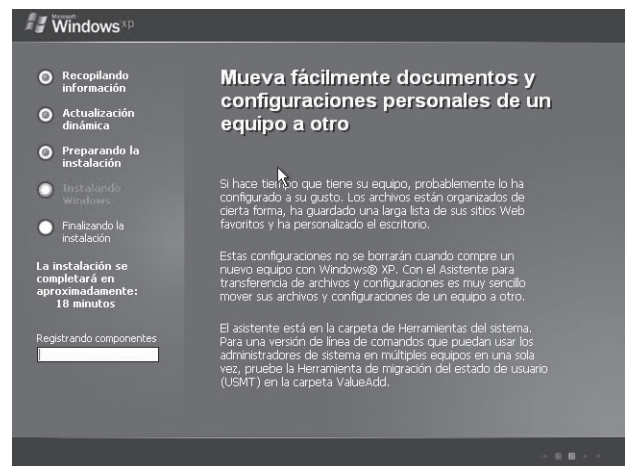


Figura 2.24

ACTIVIDADES 2.1



- Instale Windows XP en su ordenador o en una máquina virtual. Haga las capturas que crea necesarias para ilustrar el proceso de instalación.

2.2.2 INSTALACIÓN DE WINDOWS 7

Los requerimientos mínimos para instalar Windows 7 son: procesador de 32 ó 64 bits con frecuencia superior a 1 GHz, 1 GB de memoria RAM para la versión de 32 bits, 2 GB para 64 Bits, 16 GB de espacio libre en disco duro, 20 para 64 bits y una placa de vídeo con soporte para DirectX 9.

Antes de instalar Windows 7, es recomendable realizar una copia de seguridad de sus archivos, ya que al borrar la información del disco duro, no podrá recuperar jamás estos archivos.

La instalación de Windows 7 es más sencilla que la de Windows XP, ya que en este sistema no existe la instalación en modo texto que tenía XP.

Al igual que desde XP, partimos desde “Pulse una tecla para iniciar desde CD/DVD”, pulsamos una tecla, nos aparecerán una serie de pantallas de carga de nuestro sistema operativo, puede ser que haya momentos en los que la pantalla se quede en negro completamente, es normal, no hay que preocuparse.

Una vez ha terminado de cargar la instalación (Figura 2.25) nos aparecerá una pantalla donde nos va a preguntar el idioma, el formato de hora y moneda y la distribución del teclado que vamos a utilizar. Elegimos **Español**, si no viniese por defecto (Figura 2.26).

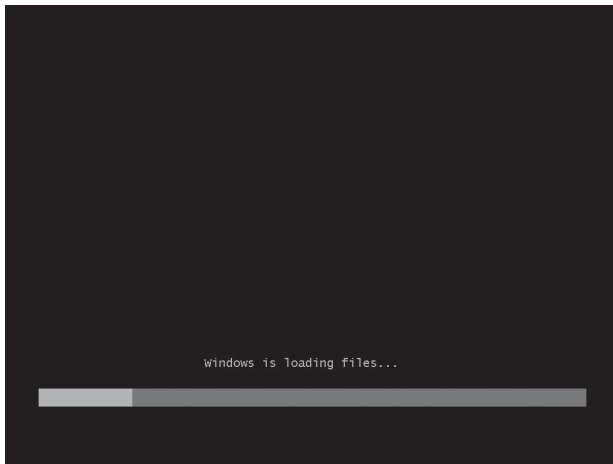


Figura 2.25



Figura 2.26

Acto seguido podemos iniciar la instalación. Aquí también podremos reparar una instalación anterior y consultar algunos aspectos básicos antes de iniciar la instalación de Windows 7. En nuestro caso, como es una instalación limpia, pulsaremos sobre **Instalar ahora** (Figura 2.27).

Tras esto nos aparecerá el acuerdo de licencia de Microsoft. Una vez leído, pulsamos sobre **Aceptar** y continuamos la instalación (Figura 2.28).



Figura 2.27

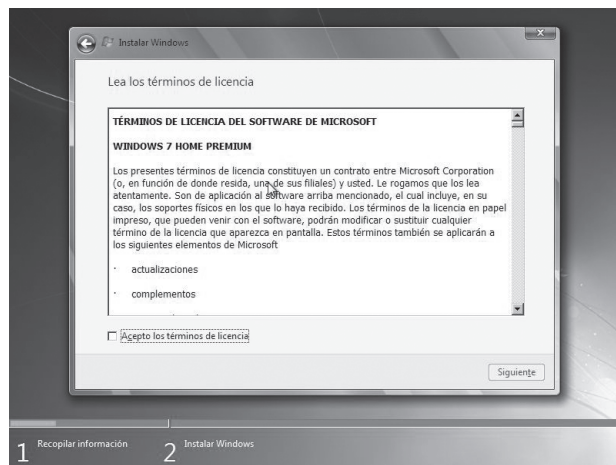


Figura 2.28

Estamos en la pantalla del tipo de instalación. Si tenemos algún sistema operativo de Microsoft anterior a 7 podemos actualizar dicho sistema, manteniendo nuestra información. En caso contrario debemos elegir la instalación personalizada (Figura 2.29).

En la siguiente pantalla nos aparecerán las unidades físicas y lógicas donde podemos instalar nuestro Windows 7. Igual que sucedía en Windows XP, hay que tener cuidado con la unidad que seleccionamos, pues podríamos perder información importante para nosotros (Figura 2.30).

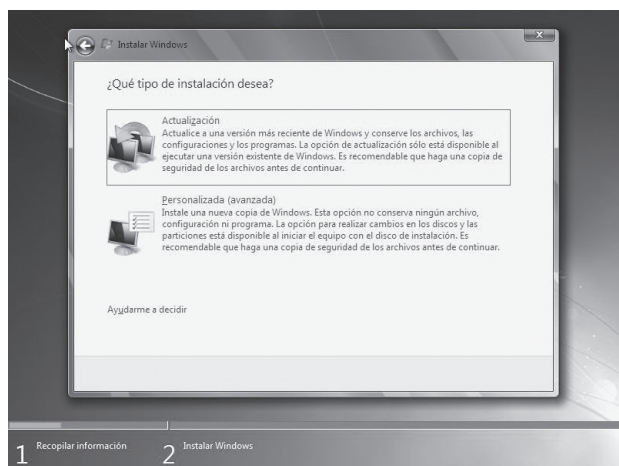


Figura 2.29

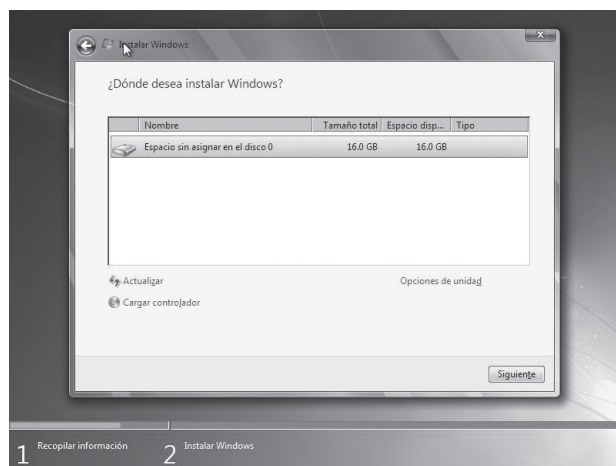
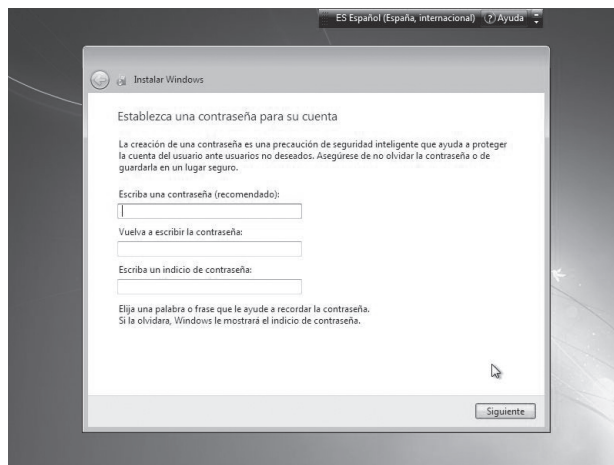


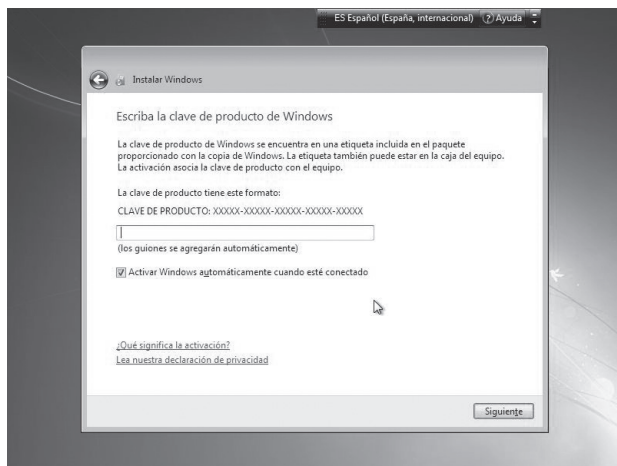
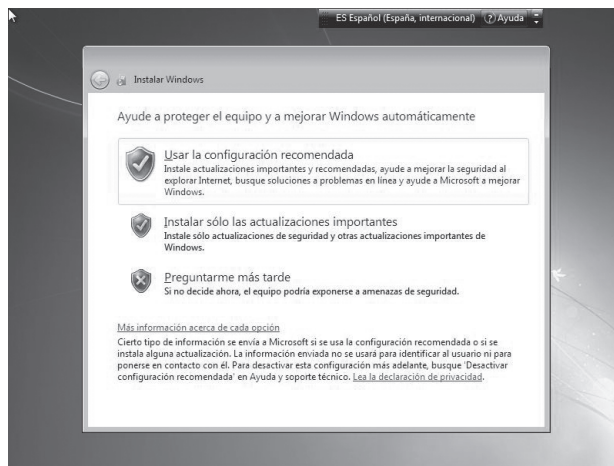
Figura 2.30

En las siguientes pantallas debemos elegir el nombre del usuario y del equipo en la red, así como la contraseña del usuario que hemos creado en la pantalla anterior (Figura 2.31). De nuevo, si dejamos la contraseña en blanco no será necesario introducir contraseña alguna para iniciar el ordenador (Figura 2.32).

**Figura 2.31****Figura 2.32**

Ahora debemos introducir la clave del producto, al igual que en Windows XP, debe estar en una pegatina pegada en el lateral del equipo o dentro de la caja donde estaba el CD/DVD de instalación del sistema operativo (Figura 2.33).

Después nos preguntará sobre las actualizaciones automáticas, debemos utilizar la configuración recomendada para que nuestro sistema esté siempre actualizado, ya que estas actualizaciones corregirán errores o vulnerabilidades detectadas, o incluso proporcionarán nuevas funcionalidades al sistema operativo (Figura 2.34).

**Figura 2.33****Figura 2.34**

Por último, solo nos queda establecer la hora y zona horaria, o confirmar las que nos presenta y establecer el tipo de red al que pertenecemos en este momento. Una red de hogar, de empresa o una red pública (Figura 2.35).

Hay que tener en cuenta que cuando cambiemos a una red distinta de ésta, nos volverá a preguntar a qué red pertenecemos. Un ejemplo puede ser un portátil que cuando lo tenemos en casa forma parte de una red doméstica, en el instituto a una red de trabajo y en una cafetería a una red pública. Según pertenezca a una red u otra podremos y podrán acceder o no a carpetas compartidas, impresoras en red, etc. (Figura 2.36).

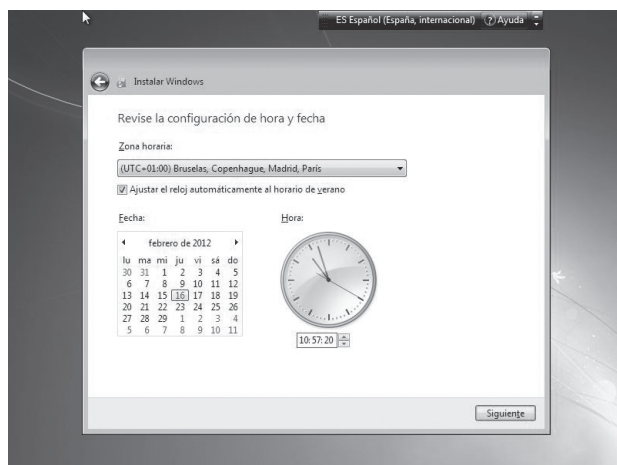


Figura 2.35



Figura 2.36

ACTIVIDADES 2.2



- Instale Windows 7 en su ordenador o en una máquina virtual. Haga las capturas que crea necesarias para ilustrar el proceso de instalación.

2.2.3 INSTALACIÓN DE UBUNTU

Ubuntu es una distribución de Linux, quizás la más conocida y utilizada en la actualidad. Los requisitos mínimos para instalar la última versión (11.10) son: Pentium 4 a 1 GHz, 64 MB (mínimo para que se instale), recomendado 512 MB y 5 GB de disco duro.

Con el tipo de instalación que se presenta a continuación se borrarán todos los archivos de nuestro ordenador, así que, en el caso de que se trate de una reinstalación, debemos guardar todos los datos que tengamos en él en soportes externos como CD, DVD o dispositivos de memoria USB, antes de comenzar la instalación.

La instalación de Ubuntu es parecida a la de Windows 7, en primer lugar se hace todo en modo gráfico, lo cual facilita bastante la instalación para usuarios poco experimentados y noveles. Una vez arrancamos con el CD/DVD de instalación cargará una serie de archivos, puede que este proceso sea un poco pesado, dependerá del equipo en el que lo instalemos y del estado del CD/DVD utilizado para la instalación.

Una vez ha cargado los archivos necesarios para la instalación o prueba (Figura 2.37). Nos aparecerá una pantalla donde debemos elegir el idioma y el modo en el que vamos a utilizar este sistema. Podemos elegir entre probar el sistema sin instalar nada, o instalar directamente la distribución en nuestra máquina. En nuestro caso he elegido la segunda opción, la de la instalación. También se nos dará la posibilidad de la instalación aunque arranquemos en modo prueba (Figura 2.38).



Figura 2.37

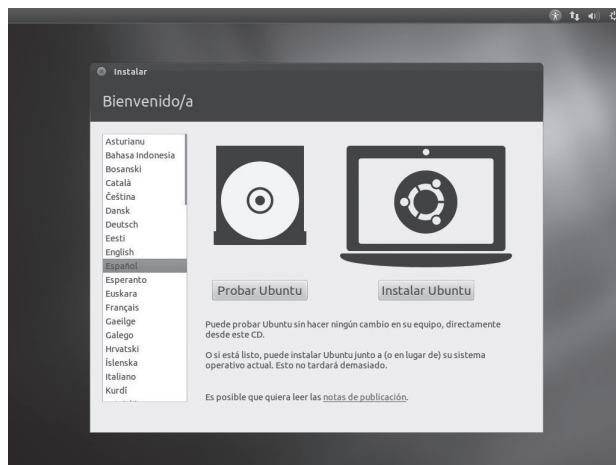


Figura 2.38

El siguiente paso comprueba que tenemos una unidad de disco con espacio suficiente para la instalación y que estamos conectados a Internet. Podemos elegir que se instalen las actualizaciones en el proceso de instalación y también elegir que se instale software de terceros, como los *codecs* para reproducción de ficheros de vídeo, audio, etc. (Figura 2.39).

En la siguiente pantalla nos va a permitir instalar Ubuntu junto a otros sistemas operativos, actualizar otra versión de Linux, borrar el disco e instalarlo en solitario, y alguna opción más. En nuestro caso, como se comentó al principio, elegiremos la opción **Borrar todo el disco e instalar** (Figura 2.40).

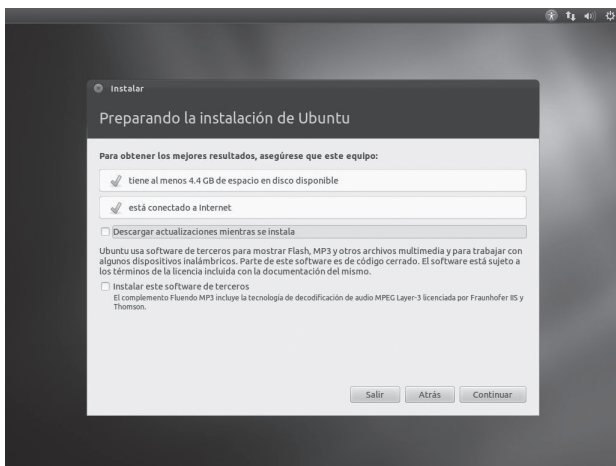


Figura 2.39

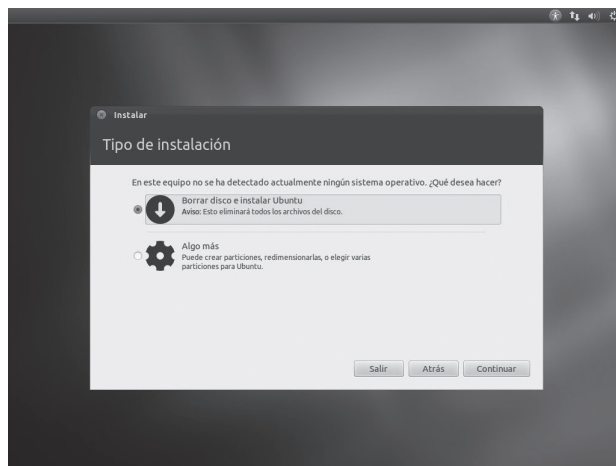


Figura 2.40

Comienza la instalación, ya solo nos queda configurar una serie de ajustes. En primer lugar la localización, podemos establecer la localización introduciendo el nombre de la población directamente o seleccionado del mapa (Figura 2.41).

A continuación, la distribución del teclado. Si hemos elegido bien la configuración en la pantalla anterior deberá aparecernos, por defecto, la configuración más adecuada para dicha localización (Figura 2.42).

En la siguiente ventana crearemos un usuario, este usuario será administrador del sistema, en este apartado podemos configurar varias opciones sobre el usuario: establecer una contraseña, que no sea necesaria dicha contraseña para entrar y cifrar la carpeta personal (Figura 2.43).

Ya esta terminada la configuración, el resto de la instalación consistirá en la copia de archivos al ordenador. Una vez finalizada dicha copia debemos pulsar sobre reiniciar y debemos extraer el CD/DVD de la unidad cuando se nos pida (Figura 2.49).

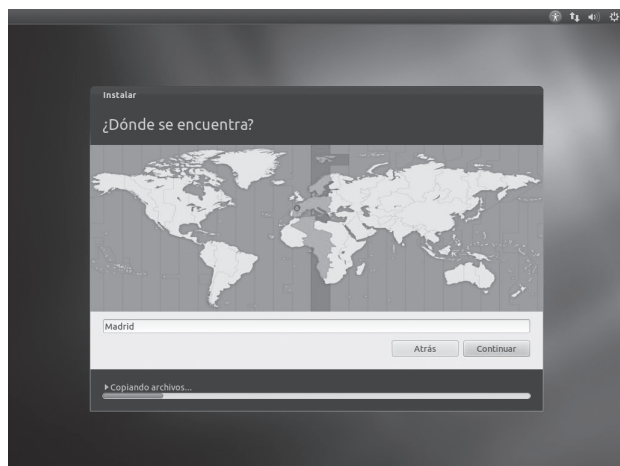


Figura 2.41

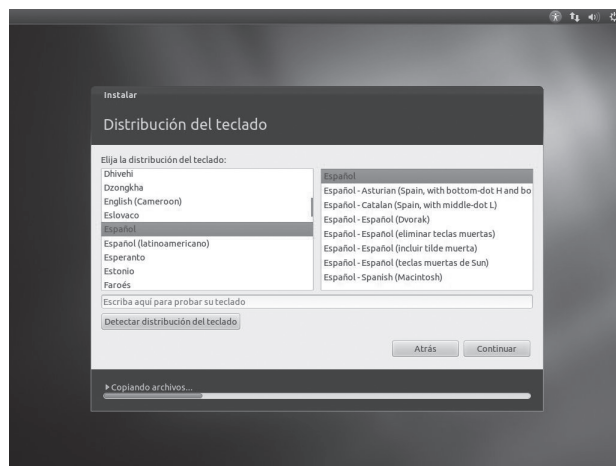


Figura 2.42

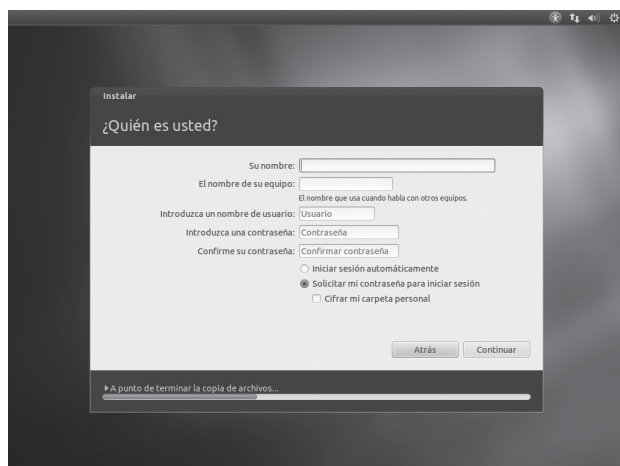


Figura 2.43



Figura 2.44

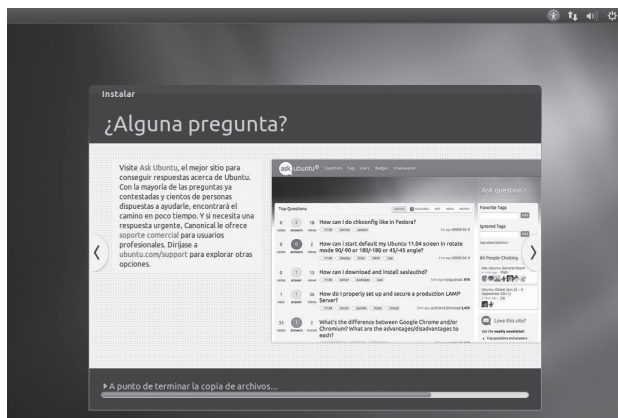


Figura 2.45



Figura 2.46

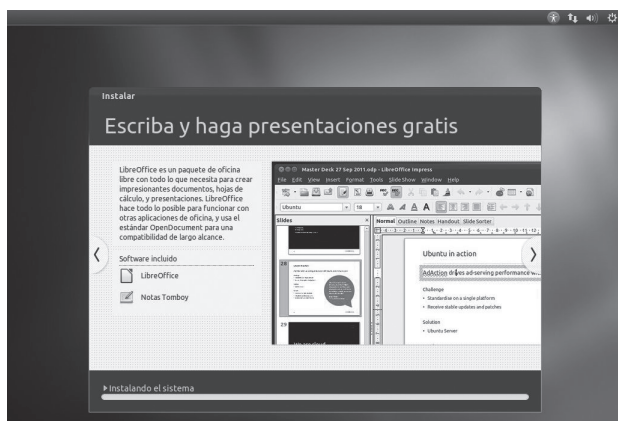


Figura 2.47



Figura 2.48

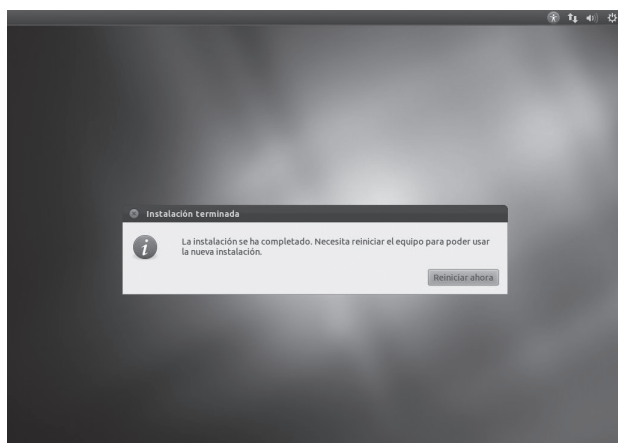


Figura 2.49

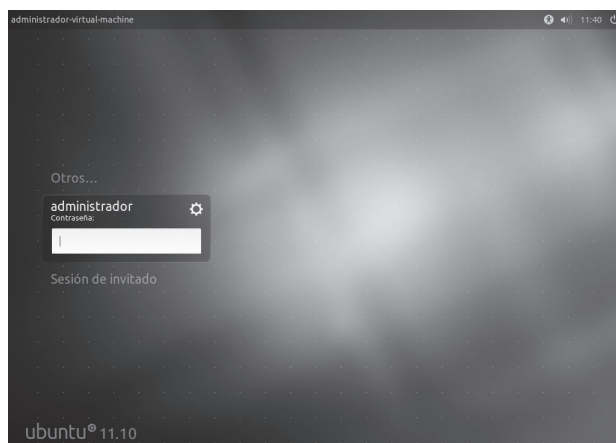


Figura 2.50

ACTIVIDADES 2.3



- Instale Ubuntu en su ordenador o en una máquina virtual. Haga las capturas que crea necesarias para ilustrar el proceso de instalación.

2.3 INSTALACIÓN DEL SERVIDOR WEB APACHE, PREPROCESADOR PHP Y SGBD MYSQL EN WINDOWS

Un servidor web no es más que un programa que está en ejecución de forma continua en un ordenador (también se utiliza el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que contesta a estas peticiones de forma adecuada, sirviendo una página web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

Instalar un servidor web en nuestro PC nos permitirá, entre otras cosas, poder montar nuestra propia página web sin necesidad de contratar *hosting*, probar nuestros desarrollos en local, acceder a los ficheros de nuestro ordenador desde un PC remoto (aunque para esto existen otras alternativas, como utilizar un servidor FTP) o utilizar alguno de los programas basados en web tan interesantes que están saliendo a la luz últimamente.

Uno de los servidores web más populares del mercado, y el más utilizado actualmente, es Apache, de código abierto y gratuito, disponible para Windows y GNU/Linux, entre otros. Su instalación es bastante sencilla



Figura 2.51. Apache

Lo primero que debemos hacer es descargar la última versión disponible de Apache 2.2, lo hacemos desde la URL: <http://httpd.apache.org/download.cgi>

En este momento la última versión es Apache 2.2.22 pero cualquier 2.2 e incluso la 2 se instala de forma similar. Como obviamente no nos interesan los fuentes ni la versión para Linux descargamos Win32 Binary including OpenSSL, que incluye también soporte para SSL, puede sernos útil más adelante.

Apache HTTP Server 2.2.22 (httpd)

The Apache HTTP Server Project is pleased to announce the release of Apache HTTP Server (httpd) version 2.2.22.

For details see the [Official Announcement](#) and the [CHANGES_2.2](#) or condensed [CHANGES_2.2.22](#) lists

Add-in modules for Apache 2.0 are not compatible with Apache 2.2. If you are running third party add-in modules, you may need previous versions. Modules compiled for Apache 2.2 should continue to work for all 2.2.x releases.

- Unix Source: <http://httpd-2.2.22.tar.gz> [PGP] [MD5] [SHA1]
- Unix Source: <http://httpd-2.2.22.tar.bz2> [PGP] [MD5] [SHA1]
- Win32 Source: <http://httpd-2.2.22-win32-src.zip> [PGP] [MD5] [SHA1]
- Win32 Binary without crypto (no mod_ssl) (MSI Installer): http://httpd-2.2.22-win32-x86-no_ssl.msi [PGP] [MD5] [SHA1]
- Win32 Binary including OpenSSL 0.9.8t (MSI Installer): <http://httpd-2.2.22-win32-x86-openssl-0.9.8t.msi> [PGP] [MD5] [SHA1]
- NetWare Binary: <http://httpd-2.2.22-netware.zip> [PGP] [MD5] [SHA1]
- [Security and official patches](#)
- [Other files](#)

Figura 2.52. Descarga apache

Tras la descarga, lanzaremos el instalador y realizamos la instalación, la instalación es muy sencilla y solo debemos configurar algunos datos.

En primer lugar, debemos introducir la información del servidor, indicando dominio, nombre del servidor, email del administrador, y si vamos a instalar Apache para todos los usuarios de nuestro Windows o solo el actual. En nuestro caso el dominio de red y el nombre de la maquina elegido para nuestro servidor ha sido *localhost* (maquina local).

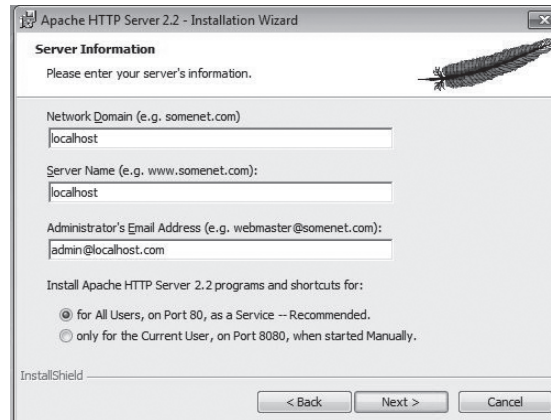


Figura 2.53. Información del servidor

Como hemos indicado antes utilizaremos la instalación personalizada (*custom*). De este modo podremos personalizar un poco la instalación. Si elegimos la opción típica nos cargará los valores por defecto de la aplicación. No le de miedo elegir esta opción, ya que la configuración personalizada es sencilla.

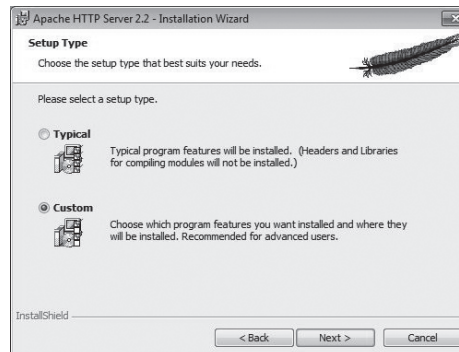


Figura 2.54. Instalación personalizada (*custom*)

En la siguiente ventana nos va a aparecer la carpeta en la que se nos va a instalar apache, en nuestro caso la voy a cambiar *c:\web*, más que nada por comodidad, para no que tener que estar accediendo a la carpeta por defecto de la instalación: *c:\Archivos de Programa\Apache Software Foundation\Apache2.2*

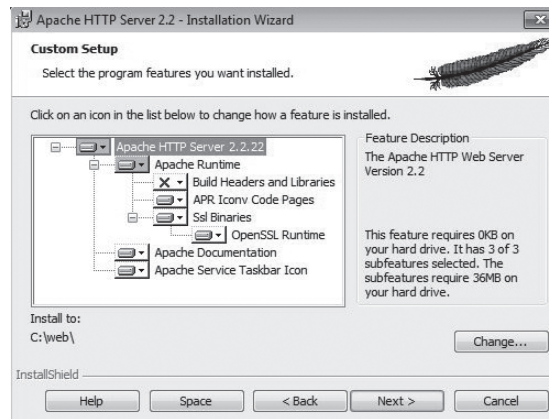


Figura 2.55. Carpeta de instalación

Ya está terminada la instalación de Apache para Windows. Pulsamos en **Finalizar** para dar por concluida la instalación.



Figura 2.56. Instalación finalizada

Para comprobar que Apache se ha instalado y está funcionando nos vamos al símbolo de sistema y escribimos: `c:\web\httpd.exe -k install` y presionamos **Intro**, debería aparecer un mensaje indicando que apache está instalado (Figura 2.57).

Para verificar el funcionamiento del servidor en nuestro navegador, entramos a la URL: `http://localhost` nos mostraría una página web con el mensaje “It works!”(Figura 2.58).

```
C:\Users\usuario.usuario-PC>c:\web\bin\httpd -k install
[error] Apache2.2: Service is already installed.
```

Figura 2.57. Comprobación de la instalación



Figura 2.58. It Works

2.3.1 CONFIGURACIÓN BÁSICA DE APACHE

En principio está todo hecho, pero en algunos casos es necesario modificar y personalizar el funcionamiento de Apache, para ello vamos a editar **httpd.conf** que se encuentra en dentro de `c:\web\conf`.

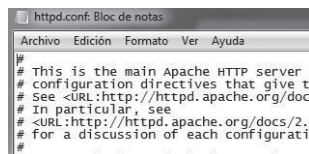


Figura 2.59. httpd.conf

- **ServerRoot:** define la ruta donde se encuentra Apache2.2, en nuestro caso debe ser `c:\web`.
- **ServerName:** nos permite definir el nombre del servidor y un puerto determinado.

```
#
ServerRoot "C:/web"
```

Figura 2.60. ServerRoot

```
#
#ServerName localhost:80
```

Figura 2.61. ServerName

- **DocumentRoot:** en este directorio irán alojados los ficheros de nuestra/s página/s web. Si cambiamos la ubicación de DocumentRoot, no cambia la dirección de acceso de nuestro navegador `http://localhost` (Figura 2.62).
- **Listen:** puerto por el que Apache se comunica con el exterior, normalmente será el puerto 80 (Figura 2.63).

```
"
DocumentRoot "C:/web/htdocs"
```

Figura 2.62. DocumentRoot

```
#Listen 12.34.56.78:80
Listen 80
```

Figura 2.63. Listen

Se han escrito libros enteros sobre la configuración y uso de Apache, por lo que esto queda lejos del propósito de este capítulo. Si le interesa personalizar totalmente el servidor puede echar un vistazo a la documentación de Apache en la web de la Fundación (<http://projects.apache.org/docs/index.html>).

2.3.2 INSTALACIÓN Y CONFIGURACIÓN DEL MÓDULO PHP EN WINDOWS

Apache, por sí solo, solamente permite mostrar al visitante webs estáticas, escritas en HTML y utilizar lenguajes que se interpreten del lado del cliente, como JavaScript.

Si queremos crear una web dinámica, con la que el usuario pueda interactuar realmente y que modifique su comportamiento según esta interacción, tendremos que utilizar un lenguaje de programación como PHP, Perl, JSP o ASP.

El código PHP se puede incluir en un documento HTML, lo que nos permite crear páginas web dinámicas. Estas líneas de código serán interpretadas en el servidor y el resultado mandado al cliente después. Esto es lo que se llama *server-side* o del lado del servidor, en comparación con *client-side* o del lado del cliente, modelo en el cual es el propio PC cliente el que tiene que interpretar el código.

Lo primero que tenemos que hacer para instalar PHP es descargar los binarios de PHP. Para ellos nos dirigimos a <http://windows.php.net/download/> y descargamos el fichero zip. No utilice el instalador.



Figura 2.64. Descarga de PHP

Una vez descargado, pasamos a descomprimirlo. A efectos prácticos el directorio donde lo instalemos no importa, para comodidad vamos a suponer que lo hemos descomprimido en `c:\php`.

Antes de pasar a configurar PHP y el servidor Apache para que utilice el intérprete instalado, modificaremos la variable de entorno Path. Nos vamos a **Propiedades del sistema**. Seleccionamos la pestaña **Opciones avanzadas** y una vez ahí, **Variables de entorno**. En el cuadro de variables de sistema haremos doble clic sobre la variable Path para editarla y añadiremos al final de la lista `;c:\php` (cada entrada se separa con un punto y coma).



Figura 2.65. Equipo, botón derecho, Propiedades

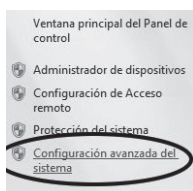


Figura 2.66. Configuración avanzada sistema

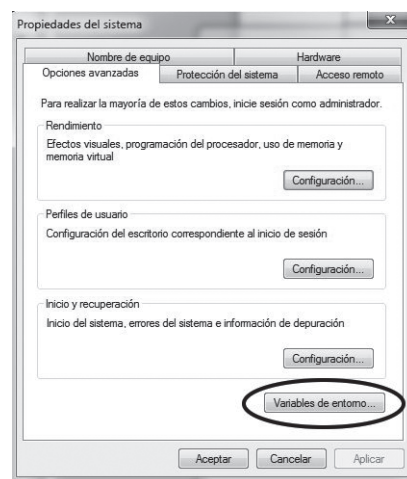


Figura 2.67. Variables de entorno

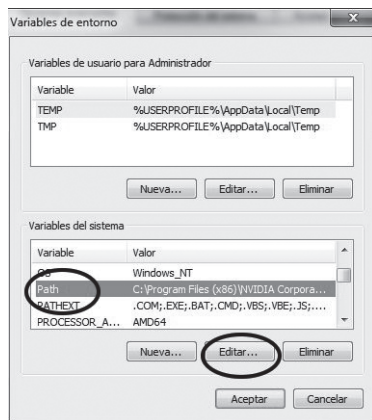


Figura 2.68. Editar variable Path

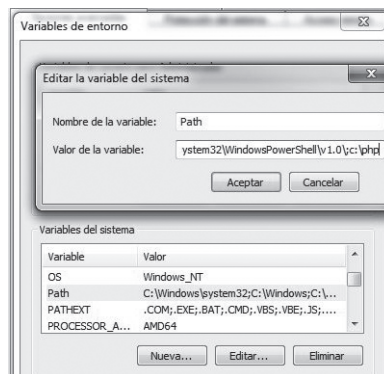


Figura 2.69. Añadir c:\php

Las distintas opciones de PHP se configuran mediante un archivo llamado **php.ini** situado en *c:\php*. Por defecto este archivo no existe, sino que tendremos que crearlo renombrando el archivo **php.ini-production**. Pasemos a editar *php.ini*.

Busque el texto **include_path = ".;c:\php\includes"** y descoméntelo (borre el ; al inicio de la línea).

Modificamos el valor de **doc_root** para que coincida con el directorio donde se guardan los archivos que mostrará Apache (por defecto, si no lo cambiamos en *httpd.conf*, este será **c:\web\htdocs**).

```

; windows: "\path1;\path2"
include_path = ".;c:\php\includes"

```

Figura 2.70. include_path

```

; cgi.force_redirect configuration below
; http://php.net/doc-root
doc_root =c:\web\htdocs

; The directory under which PHP opens the scri
; if nonemotv.

```

Figura 2.71. doc_root

Modificamos el valor de *session.cookie_path* por *session.cookie_path=*

Y, por último, modificamos el valor de *session.save_path* para poder trabajar con sesiones (el directorio debe existir, PHP no lo creará por nosotros): *session.save_path = "c:\php\siones"*.

```

; http://php.net/session.
session.cookie_path = \

```

Figura 2.72. cookie_path

```

; http://php.net/session.save-path
session.save_path = "c:\php\siones"

```

Figura 2.73. save_path

Solo falta activar algunas extensiones útiles para que estas se carguen automáticamente. Tendremos que indicar a PHP dónde se guardan las extensiones, modificando **extension_dir** de forma que contenga la ruta al directorio *ext* dentro de la carpeta donde se instaló PHP (**c:\php\ext**).

```
; On windows:
extension_dir = "c:\php\ext"
```

Figura 2.74. extension_dir

Una vez hecho esto buscamos la sección Windows Extensions que muestra una lista de distintas extensiones que podemos activar. Para cada extensión que queramos, no tenemos más que borrar el punto y coma al principio de la línea o bien añadir una nueva línea si no existe.

Necesitaremos activar, al menos, **php_MySQL.dll** para que PHP pueda guardar y obtener datos de una base de datos MySQL. También es muy recomendable **php_MySQLi.dll**, que ofrece conexiones mejoradas, y por ultimo **php_mbstring.dll**, necesaria para utilizar codificaciones de caracteres en las que cada carácter ocupa más de un byte.

```
;extension=php_ldap.dll
extension=php_mbstring.dll
;extension=php_exif.dll
extension=php_mysql.dll
extension=php_mysqli.dll
;extension=php_oci8.dll
```

Figura 2.75. Windows Extensions

Es recomendable no activar una extensión si no conoce cual es su función, entre otras cosas porque muchas de ellas necesitan otros archivos, y en el caso de que PHP no los encuentre se quejará.

Una vez configurado PHP es el momento de modificar la configuración de Apache para trabajar con este. El archivo de configuración de Apache es **httpd.conf**, situado por defecto en **C:\Archivos de programa\Apache Group\Apache2\conf**, hay que tener en cuenta que nosotros hemos instalado apache en **c:\web** por tanto la carpeta de configuración será **c:\web\conf**.

Busque el texto que viene en la Figura 2.76, inmediatamente después encontrará una lista de módulos que se cargarán con Apache.

Añada al final de la lista **LoadModule php5_module "c:/php/php5.dll"** para cargar el módulo de PHP 5.

```
# Directives contained here are used
# to statically compiled modules (those that
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo
#
LoadModule actions_module modules/mod_actions
```

Figura 2.76. LoadModule

```
#LoadModule usertrack_module modules/mod_usertrack
#LoadModule version_module modules/mod_version
#LoadModule vhost_alias_module modules/mod_vhost_alias
LoadModule php5_module "c:/php/php5.dll"
```

Figura 2.77. Carga del modulo de PHP

Busque el texto que viene en la Figura 2.78. Aparece una lista de tipos MIME que el servidor es capaz de reconocer.

Añada al final de la lista **AddType application/x-httpd-php .php**.

```
# probably should define those extensions
#
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-httpd-php .php
#
# AddHandler allows you to map certain
```

Figura 2.78. MIME

```
# If the AddEncoding directives above
# probably should define those extensions
#
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
```

Figura 2.79. MIME

Busque **DirectoryIndex index.html**, como aparece en la Figura 2.80 y sustituya por **DirectoryIndex index.php index.html**. Esto indicará a Apache que los archivos *php* también pueden utilizarse como archivos de índice de un directorio. Añada al final del fichero la línea **PHPIniDir c:\php** para indicar al servidor web dónde encontrar el archivo *php.ini* y reinicie el PC.

```
#
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

Figura 2.80. DirectoryIndex

```
#
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

Figura 2.81. DirectoryIndex

```
SSLRandomSeed connect
</IfModule>
PHPIniDir "c:\php"
```

Figura 2.82. PHPIniDir

Hora de probar que PHP se ha instalado correctamente. Esto se suele hacer creando un archivo *php* (crear un nuevo archivo con cualquier editor de textos, con nombre, por ejemplo *info.php*) cuyo contenido será:

```
< ?
phpinfo();
?>
```

Introduzca en la barra de direcciones del navegador la URL: *http://localhost/info.php*

PHP Version 5.3.8

System	Windows NT ADMIN1 6.1 build 7600 (Windows 7 Business Edition) i586
Build Date	Aug 23 2011 11:47:20
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--disable-isapi" "--without-mssql" "--without-pdo-mssql" "--without-p3web" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10sdk\shared" "--with-oci8=D:\php-sdk\oracle\instantclient10sdk\shared" "--with-oci8-11g=D:\php-sdk\oracle\instantclient111sdk\shared" "--enable-object-out-dir=.obj/" "--enable-com-dotnet" "--with-mcrypt=static" "--disable-static-analyze"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini

Figura 2.83. Comprobación de la instalación de PHP

2.3.3 INSTALACIÓN DE MYSQL EN WINDOWS

Instalar MySQL, dependiendo de si nuestro sistema operativo es de 32 ó 64 bits instalaremos una versión u otra de MySQL.

Para descargar MySQL nos vamos a la dirección:

<http://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-5.5.15.0.msi/from/http://mysql.mirror.kangaroot.net/>

Guardamos el archivo en nuestro ordenador. Hacemos doble clic sobre el programa descargado y se inicia la instalación. En primer lugar debemos aceptar los términos de la licencia para poder continuar con la instalación.

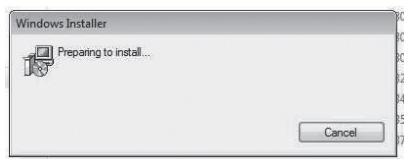


Figura 2.84. Preparando la instalación

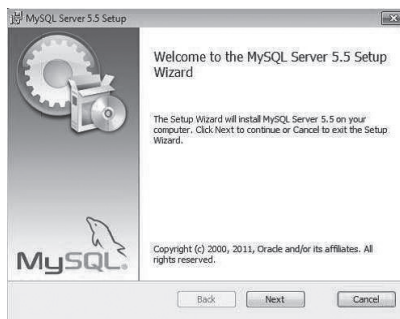


Figura 2.85. Bienvenido

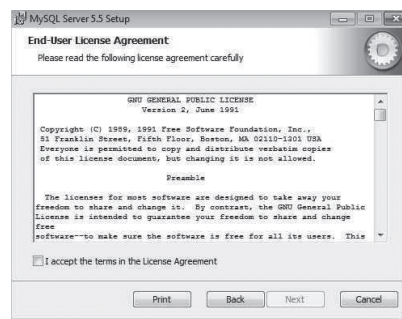


Figura 2.86. Acuerdo de licencia

Una vez aceptados los términos nos preguntará que tipo de instalación vamos a efectuar. En nuestro caso con la instalación **típica** nos será más que suficiente, por tanto, pulsamos sobre el botón **Typical**. Ya estamos preparados para instalar MySQL. Durante la instalación nos aparecerán una serie de pantallas en las que simplemente debemos dar a siguiente (**Next**) para continuar. Una vez ha finalizado la instalación pulsamos sobre **Finish** para que nos aparezca el Asistente de configuración de MySQL (**MySQL Instance Configuration Wizard**).

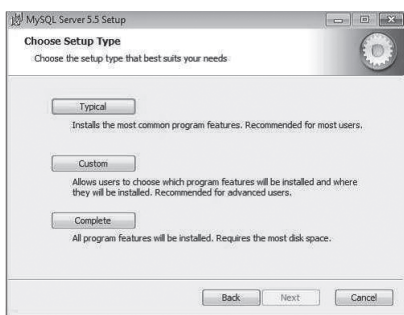


Figura 2.87. Tipo de instalación

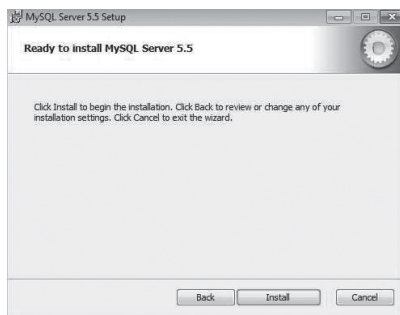


Figura 2.88. Listo para instalar

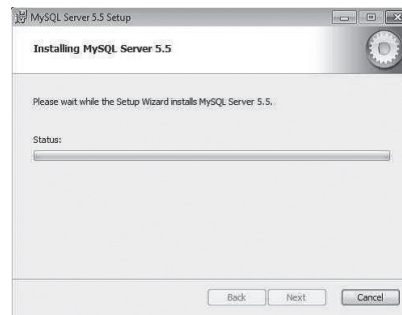


Figura 2.89. Instalando



Figura 2.90. Next

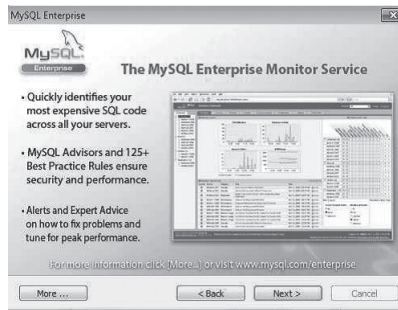


Figura 2.91. Next

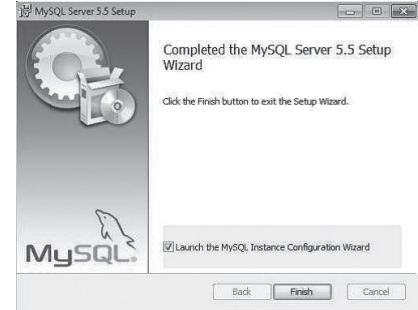


Figura 2.92. Finalizar

Estamos ahora frente al asistente de Configuración de MySQL. En primer lugar debemos elegir entre Configuración Detallada (**Detailed Configuration**) o Configuración Estandar (**Standard Configuration**). En nuestro caso hemos seleccionado la configuración detallada, pues podemos personalizar un poco más MySQL. Tras esto y si hemos elegido la anterior configuración, nos preguntara que tipo de maquina vamos a utilizar para instalar MySQL. En nuestro caso hemos seleccionado **Developer Machine**, ya que nos permitirá ejecutar MySQL y no consumirá excesiva memoria, cosa que no ocurre con el resto de instancias.

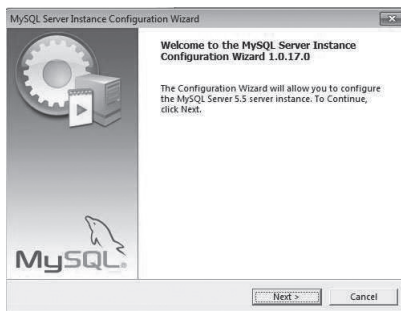


Figura 2.93. Configuración del servidor

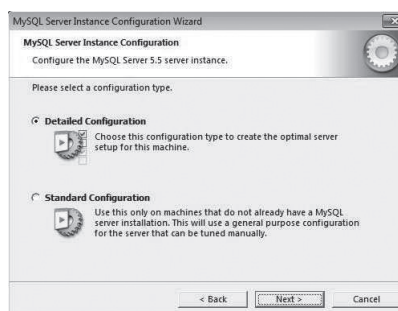


Figura 2.94. Tipo de configuración



Figura 2.95. Tipo de servidor

En esta ocasión debemos seleccionar el uso de la base de datos, hay tres tipos: **Multifunctional**, **Transactional** y **Non-transactional**, igual que anteriormente, dependiendo del uso que vayamos a dar al sistema gestor escogeremos una u otra opción. En nuestro caso, como las bases de datos que vamos a utilizar son de propósito general, y van a ser de tipo InnoDB (que nos permite el uso de claves foráneas y otras cosas) hemos seleccionado **Multifunctional Database** (Figura 2.96).

En la siguiente pantalla debemos elegir la ruta de instalación de las tablas InnoDB en caso de no querer utilizar la configuración por defecto, en principio no debemos hacer nada en especial en esta pantalla, simplemente pulsamos **Next**.

Tras esta debemos seleccionar el número aproximado de usuarios que accederán de forma concurrente a la base de datos. En nuestro caso hemos seleccionado **Decision Support (DDS)/OLAP**, esta configuración esta concebida para aplicaciones de bases de datos que no requieren un número alto de conexiones concurrentes.

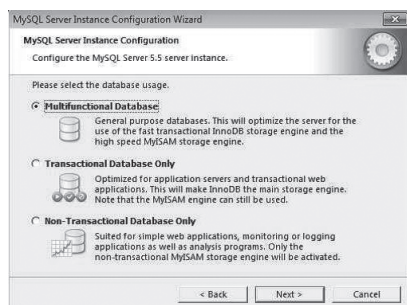


Figura 2.96. Uso de la base de datos

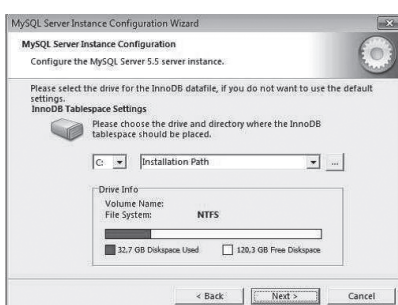


Figura 2.97. InnoDB tablespace

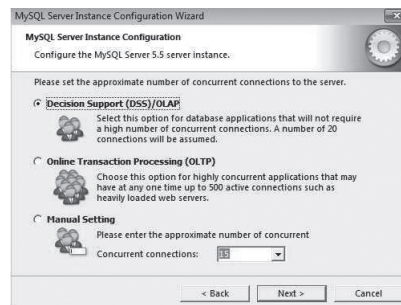


Figura 2.98. Número de conexiones

En este punto llegamos a las opciones de red (**Networking options**), habilitaremos (o no) la red TCP/IP y estableceremos un puerto mediante el cual se realizarán la conexiones (incluso podemos agregar una excepción al cortafuegos de ese puerto seleccionado). Si deshabilitamos esta opción, el acceso a la bases de datos solo podrá hacerse de forma local (desde la propia máquina).

En la siguiente pantalla debemos seleccionar el conjunto de caracteres por defecto (**default character set**), como podemos ver en la Figura 2.101 hemos seleccionado **Standard Character Set**, ya que es el más apropiado para lenguajes de Europa Occidental (**West European languages**).

En la siguiente pantalla (Figura 2.101) nos da la opción de instalar MySQL como un servicio de Windows (**Install As Windows Service**), esto implica que estará siempre en funcionamiento. Si esta opción no la marcamos, cada vez que queramos utilizar el gestor deberemos arrancarlo de forma manual.

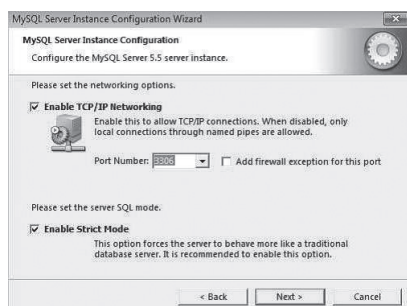


Figura 2.99. TCP/IP Networking



Figura 2.100. default carácter set



Figura 2.101. Windows options

Ahora debemos establecer las opciones de seguridad (**security options**). En esta pantalla estableceremos la contraseña para el usuario root (administrador) del sistema gestor, nos da la posibilidad de que la cuenta sea anónima, es decir sin contraseña, aunque el mismo nos advierte que nos quedará un sistema inseguro.

Ya hemos terminado la configuración, tan solo debemos pulsar sobre **Execute** y se concluirá la instalación de MySQL en nuestro equipo, una vez ha finalizado, ya se ha completado la instalación, por tanto podremos pulsar sobre **Finish** para dar por concluida la instalación.



Figura 2.102. Security options

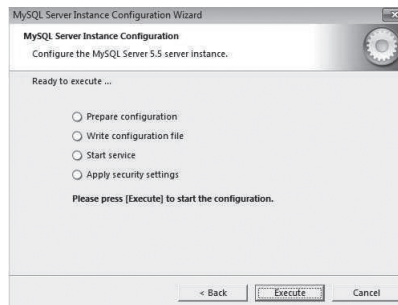


Figura 2.103. Execute

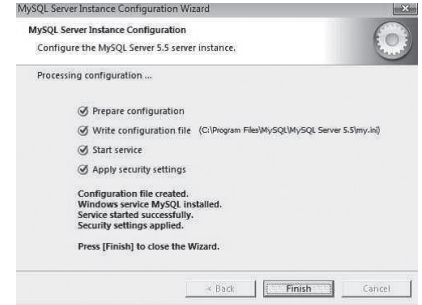


Figura 2.104. Finish

Para poder utilizar MySQL en nuestro ordenador debemos pulsar sobre *Inicio* → *MySQL* → *MySQL Command Line Client*.

Nos pedirá la contraseña que hemos puesto anteriormente (Figura 2.102) y ya podemos utilizar dicho gestor.

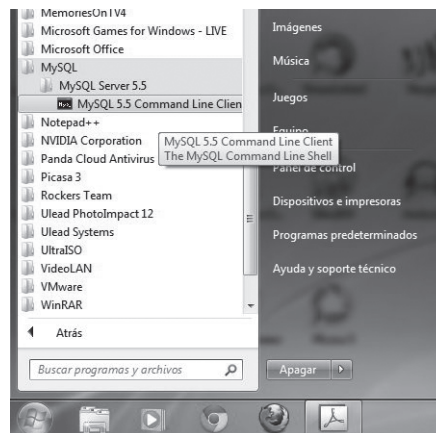


Figura 2.105. Ejecutar

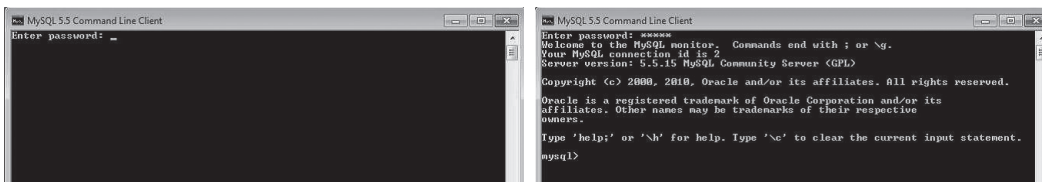


Figura 2.106. Enter password

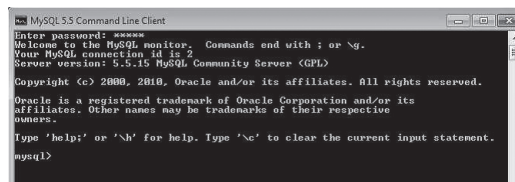


Figura 2.107. MySQL

ACTIVIDADES 2.4

➔ Descargue, instale y configure en su ordenador o en una máquina virtual con Windows: Apache, PHP y MySQL.

2.4 INSTALACIÓN DEL SERVIDOR WEB APACHE, PREPROCESADOR PHP Y SGBD MYSQL EN UBUNTU

La instalación se hará desde la consola, ya que muchas veces es el único medio disponible en servidores o cuando se hacen instalaciones mediante el uso de SSH. También hay formas más sencillas de hacerlo mediante el uso de las herramientas gráficas, como pueden ser Synaptic, KPackageKit y muchas otras utilidades.

Como hemos indicado en el título estos comandos son para Ubuntu, aunque son extensibles para la mayoría de las distribuciones basadas en Debian. Por lo tanto, nos utilizaremos el comando *sudo* para ganar privilegios de administrador y la utilidad *apt-get install* para realizar la instalación. El proceso es bastante sencillo y rápido desde la consola.

```
sudo apt-get install mysql-server-5.1 apache2 php5 php5-MySQL libapache2-mod-auth-mysql
```

```
administrador@administrador-virtual-machine:~$ sudo apt-get install mysql-server
-5.1 apache2 php5 php5-mysql libapache2-mod-auth-mysql
[sudo] password for administrador:
Leyendo listas de paquetes... Hecho
Creando árbol de dependencias
```

Figura 2.108. apt-get

```
0 actualizados, 24 se instalarán, 0 para eliminar y 62 no actualizados.
Necesito descargar 29,2 MB de archivos.
Se utilizarán 74,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? █
```

Figura 2.109. ¿Desea Continuar?

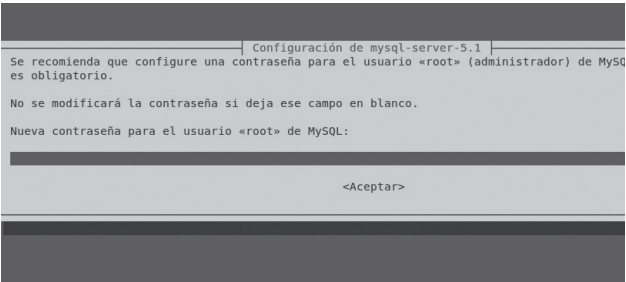


Figura 2.110. Contraseña root

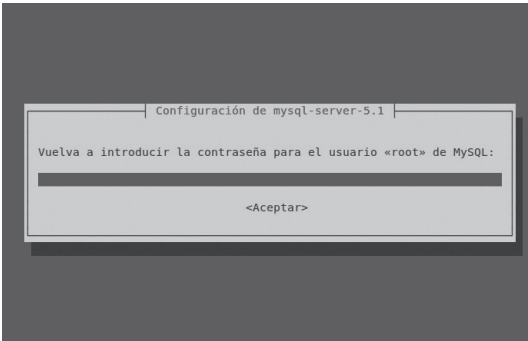


Figura 2.111. Repita contraseña root

```

Desempaquetando apache2-utils (de .../apache2-utils_2.2.20-1ubuntu1.2_i386.deb) ...
Seleccionando el paquete apache2.2-common previamente no seleccionado.
Desempaquetando apache2.2-common (de .../apache2.2-common_2.2.20-1ubuntu1.2_i386.deb) ...
Seleccionando el paquete apache2-mpm-prefork previamente no seleccionado.
Desempaquetando apache2-mpm-prefork (de .../apache2-mpm-prefork_2.2.20-1ubuntu1.2_i386.deb) ...
Seleccionando el paquete apache2 previamente no seleccionado.
Desempaquetando apache2 (de .../apache2_2.2.20-1ubuntu1.2_i386.deb) ...
Seleccionando el paquete libapache2-mod-auth-mysql previamente no seleccionado.
Desempaquetando libapache2-mod-auth-mysql (de .../libapache2-mod-auth-mysql_4.3.9-13ubuntu1_i386.deb) ...
Seleccionando el paquete php5-common previamente no seleccionado.
Desempaquetando php5-common (de .../php5-common_5.3.6-13ubuntu3.6_i386.deb) ...
Seleccionando el paquete libapache2-mod-php5 previamente no seleccionado.
Desempaquetando libapache2-mod-php5 (de .../libapache2-mod-php5_5.3.6-13ubuntu3.6_i386.deb) ...
Seleccionando el paquete php5 previamente no seleccionado.
Desempaquetando php5 (de .../php5_5.3.6-13ubuntu3.6_all.deb) ...
Seleccionando el paquete php5-mysql previamente no seleccionado.
Desempaquetando php5-mysql (de .../php5-mysql_5.3.6-13ubuntu3.6_i386.deb) ...
Procesando disparadores para ureadahead ...
ureadahead will be reprofiled on next reboot
Procesando disparadores para man-db ...
Procesando disparadores para ufw ...
Configurando libnet-daemon-perl (0.48-1) ...
Configurando liblprc-perl (0.2020-2) ...

```

Figura 2.112. Instalando y configurando

```

Enabling module reqtimeout.
Configurando apache2-mpm-prefork (2.2.20-1ubuntu1.2) ...
 * Starting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
[ OK ]

Configurando apache2 (2.2.20-1ubuntu1.2) ...
Configurando libapache2-mod-auth-mysql (4.3.9-13ubuntu1) ...
Configurando php5-common (5.3.6-13ubuntu3.6) ...
Configurando libapache2-mod-php5 (5.3.6-13ubuntu3.6) ...

Creating config file /etc/php5/apache2/php.ini with new version
 * Reloading web server config apache2
apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
[ OK ]

Configurando php5 (5.3.6-13ubuntu3.6) ...
Configurando php5-mysql (5.3.6-13ubuntu3.6) ...
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
administrador@administrador-virtual-machine:~$

```

Figura 2.113. Fin de la instalación

Para asegurarnos reiniciamos MySQL y Apache con los comandos:

```

sudo /etc/init.d/apache2 restart
sudo service mysql restart

```

```

administrador@administrador-virtual-machine:~$ sudo /etc/init.d/apache2 restart
[sudo] password for administrador:
 * Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name, usir

```

Figura 2.114. apache2 restart

```

administrador@administrador-virtual-machine:~$ sudo service mysql restart
mysql start/running, process 4494
administrador@administrador-virtual-machine:~$ █

```

Figura 2.115. mysql restart

Cambiamos los permisos en la carpeta `/var/www/` para poder acceder a ella:

```
sudo chmod -R 755 /var/www
sudo chown -R www-data\: /var/www/
```

Para verificar que todo quedo bien instalado pondremos un archivo llamado `info.php` en la carpeta `/var/www/` con el siguiente contenido:

```
<?
phpinfo();
?>
```

Abrimos el navegador de Internet y navegamos la URL:

```
http://localhost/info.php
```

Deberíamos ver algo parecido a la Figura 2.116. En concreto, debemos verificar que PHP tenga soporte para MySQL. En la zona media tenemos que encontrar un bloque de MySQL parecido al de la Figura 2.117.



Figura 2.116. PHP en Ubuntu

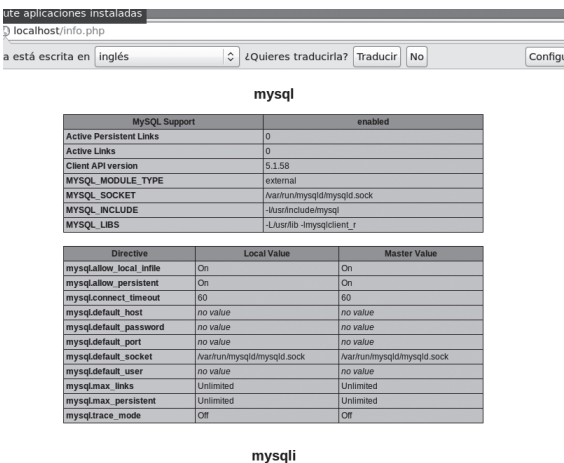


Figura 2.117. MySQL en Ubuntu

ACTIVIDADES 2.5



➤ Descargue, instale y configure en su ordenador o en una máquina virtual con Ubuntu: Apache, PHP y MySQL.

2.5 PHPMYADMIN

Se trata de una herramienta escrita íntegramente en PHP, realizada con el fin de facilitar las tareas de administración. Permite administrar de forma completa un servidor MySQL (requiere de una cuenta superusuario), así como bases de datos simples de un usuario en particular.

Las características principales de PHPMyAdmin son: crear y eliminar bases de datos. Crear, copiar, eliminar, renombrar y alternar tablas. Eliminar, editar y agregar campos, ejecutar sentencias SQL, administrar campos claves, cargar archivos de textos a las tablas, exportar los datos a diferentes formatos, administrar usuarios MySQL y privilegios, administrar múltiples servidores, crear archivos PDF del diseño de la base de datos, búsquedas globales y, además, comunicarlo en 47 idiomas diferentes.

- **Requisitos.** Es necesario contar con un servidor web con soporte de PHP, un motor de bases de datos MySQL y, por supuesto, un navegador web.
- **Instalación.** La herramienta se descarga desde sitio web oficial: *www.phpmyadmin.net*. Una vez descargado, hay que descomprimirlo en el directorio raíz del servidor web.

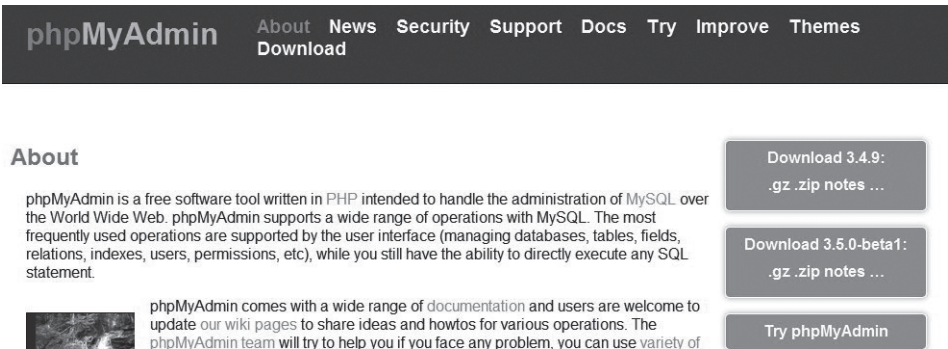


Figura 2.118. página web de phpMyAdmin



Figura 2.119. descarga de phpMyAdmin

- **Configuración.** Abrimos el archivo *config.inc.php* con algún editor de texto, como el Notepad de Windows. Debemos configurar las siguientes instancias:

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin';
```

Hay que asignar la ruta completa para acceder al PHPMyAdmin. Si trabajamos en forma local: *http://localhost/phpMyAdmin* y, en caso de trabajar en un servidor, *http://www.servidor.com/phpMyAdmin*.

```
$cfg['Servers'][$i]['host'] = 'localhost';
```

El *host* o número de IP del servidor de la base de datos MySQL.

```
$cfg['Servers'][$i]['user'] = 'usuario';
```

El nombre del usuario de la base de datos, si trabajamos en forma local, generalmente pondremos el usuario *root* (superusuario) para poder administrar usuarios y bases de datos desde el navegador. En caso de utilizar un servidor externo, solo tendremos privilegios de usuario común con una sola base de datos. Para la contraseña del usuario:

```
$cfg['Servers'][$i]['password'] = 'contraseña';
```

2.5.1 UTILIZANDO PHPMYADMIN

Configurando las líneas señaladas anteriormente ya estamos habilitados para utilizar la herramienta. En el navegador introducimos la dirección: *http://localhost/phpmyadmin*

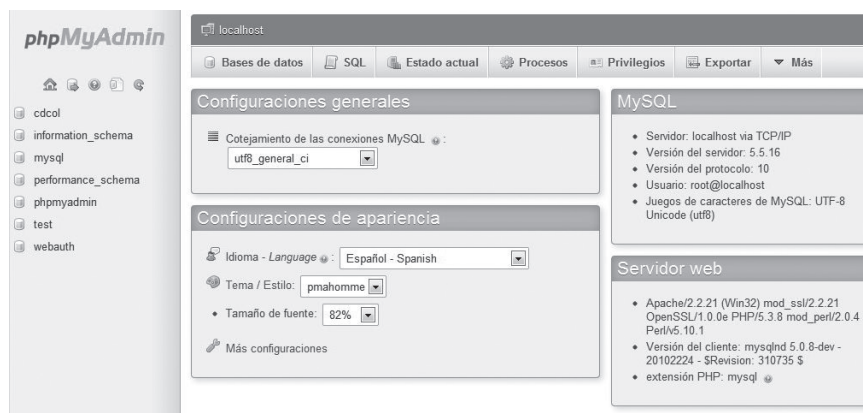


Figura 2.120. Pantalla principal de phpMyAdmin

- **Creando una base de datos.** En la página principal de phpMyAdmin, pulsamos sobre **Bases de datos** (está en la parte superior de la pantalla). Debemos completar el campo del formulario con el nombre de la nueva base de datos (pruebas) y pulsamos sobre el botón **Crear**.

Bases de datos



Figura 2.121. Crear una base de datos nueva

- **Creando una tabla.** Dentro de la base de datos recién creada, vamos a crear una tabla (*noticias*) que va a tener tres campos.

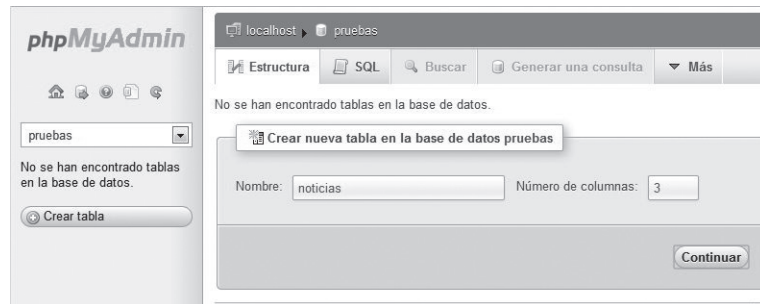


Figura 2.122. Crear una nueva tabla

- **Configurar los campos.** En el paso anterior especificamos que vamos a utilizar 3 campos, en esta página debemos configurarlos. Los campos son:

Columna	id	titulo	texto
Tipo	INT	VARCHAR	LONGTEXT
Longitud/Valores*1		250	
Predeterminado2	Ninguno	Ninguno	Ninguno
Cotejamiento			
Atributos			
Nulo	☐	☐	☐
Índice	PRIMARY	---	---
AUTO_INCREMENT	☒	☐	☐

Figura 2.123. Campos de la nueva tabla

Completados los valores, pulsamos sobre **Guardar**:

os de la tabla:			Motor de almacenamiento: ⓘ	Cotejamiento:
			InnoDB	
la PARTICIÓN: ⓘ				
Guardar O Agregar 1 columna(s) Continuar				

enum" o "set", introduzca los valores usando este formato: 'a','b','c'...
 usar una barra invertida (\) o una comilla simple (') entre esos valores; añada una barra invertida (\)...

Figura 2.124. Guardar los cambios

- **Agregando datos.** Para agregar datos hay que pulsar sobre **Insertar**, completamos el formulario (el campo *id* no es necesario completarlo ya que lo declaramos como autonumérico) y damos a **Continuar**.

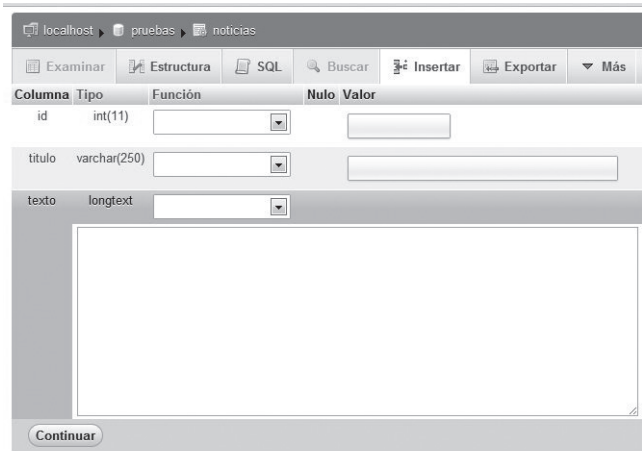


Figura 2.125. Insertar datos en la tabla

- **Observar datos.** Los datos de la tabla pueden ser visualizados pulsando sobre **Examinar**, con el cual veremos un listado de los diferentes registros contenidos en la tabla. Allí mismo, se permite la opción de **Editar** y **Borrar** los mismos.

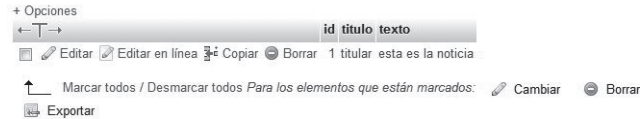


Figura 2.126. PHPMyAdmin

- **Conclusión.** PHPMyAdmin es una excelente herramienta de administración de bases de datos MySQL que permite su utilización a del navegador web. Además de ser muy intuitivo y fácil de utilizar, es realmente completo y eficaz.

ACTIVIDADES 2.6



- Descargue, instale y configure en su ordenador o en una máquina virtual con Windows la aplicación PHPMyAdmin.
- Descargue, instale y configure en su ordenador o en una máquina virtual con Ubuntu la aplicación PHPMyAdmin.

2.6 XAMPP

Existe una herramienta para instalar de forma automática lo que hemos realizado en los pasos anteriores (Apache, PHP, MySQL y PHPMyAdmin), esta herramienta se llama XAMPP.

Muchos usuarios saben por experiencia propia que la instalación de un servidor web Apache no es fácil y que se complica aún más si se desea agregar MySQL, PHP y Perl.

XAMPP es una forma fácil de instalar la distribución Apache que contiene MySQL, PHP y Perl. XAMPP es realmente simple de instalar y usar, basta con descargarlo, extraerlo y comenzar.

En este momento hay cuatro versiones de XAMPP: XAMPP para Linux, XAMPP para Windows, XAMPP para Mac OS X y XAMPP para Solaris.

XAMPP es una compilación de software libre (comparable a una distribución Linux), es gratuito y libre para ser copiado conforme a los términos de la licencia GNU General Public License. Sin embargo, solo la compilación de XAMPP está publicada bajo la licencia GPL.

2.6.1 INSTALACIÓN DE XAMPP EN WINDOWS

En primer lugar debemos de descargar XAMPP, para ello nos vamos a su página web: <http://www.apachefriends.org/es/xampp.html>, nos vamos a la sección de descargas y bajamos por la página hasta que nos aparezca **XAMPP for Windows** y pulsamos sobre la versión **Installer**, se descargará en nuestro ordenador la última versión del programa.

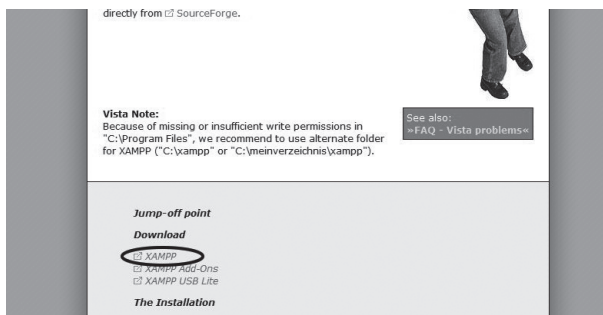


Figura 2.127. Página principal de XAMPP

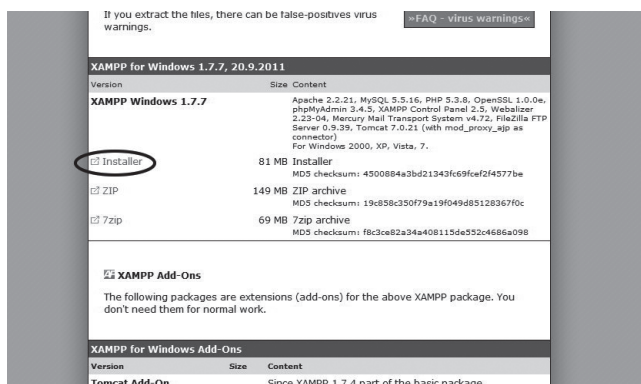


Figura 2.128. Descargar Installer

Vamos a comenzar la instalación, hacemos doble clic sobre el fichero descargado, y en primer lugar debemos elegir el idioma de instalación, en este caso no está español, aunque después si podremos elegir nuestro idioma para poder utilizar XAMPP. En nuestro caso hemos seleccionado **Inglés (English)**.

Nos da la bienvenida y nos recomienda cerrar todos los programas abiertos antes de iniciar la instalación, ya que durante la instalación pueden modificarse algunos ficheros de sistema y posiblemente tengamos que reiniciar el ordenador al finalizar la instalación.

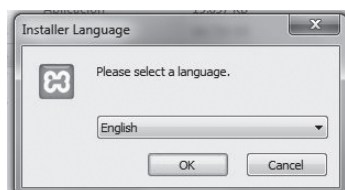


Figura 2.129. Idioma de instalación



Figura 2.130. Consideraciones iniciales

Nos pedirá ahora el directorio de instalación del programa, por defecto utilizará **c:\xampp**. Podemos cambiarlo por el que queramos, aunque el propio programa no recomienda instalarse en **Archivos de Programa** por problemas de los permisos en esta carpeta con Windows Vista y 7.

En la siguiente pantalla elegiremos opciones de la instalación de XAMPP en nuestra máquina. En primer lugar si queremos que se cree un acceso directo del programa en el escritorio. Después si queremos crear una entrada en el menú de Inicio. Y, por último, si queremos instalar Apache, MySQL y Filezilla como servicios (se ejecutarán automáticamente al iniciar Windows).

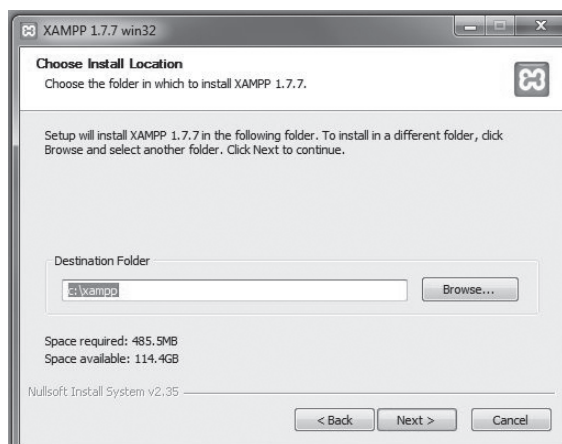


Figura 2.131. Carpeta de instalación

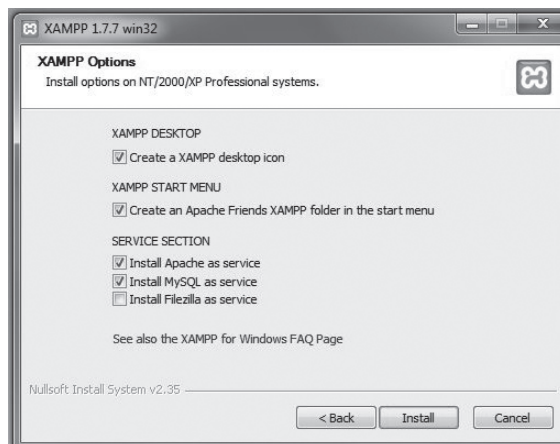


Figura 2.132. Opciones de instalación

Ya no tenemos que hacer nada más, se instalará en la carpeta seleccionada. Una vez concluida la instalación debemos pulsar en **Finish** para dar por concluida dicha instalación.

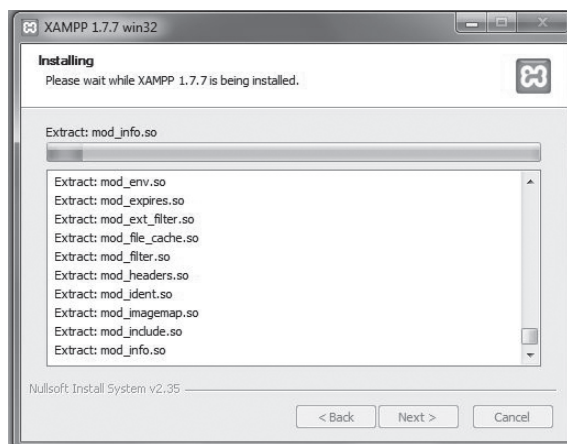


Figura 2.133. Instalación

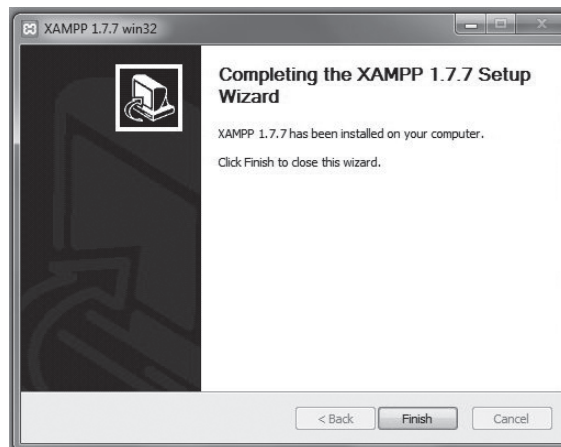


Figura 2.134. Finish

Comprueba que los puertos que utiliza tanto Apache como MySQL están libres, en este punto se abrirán una serie de ventanas de sistema que se cerraran automáticamente una vez hechas las comprobaciones pertinentes. No cierre dichas ventanas.

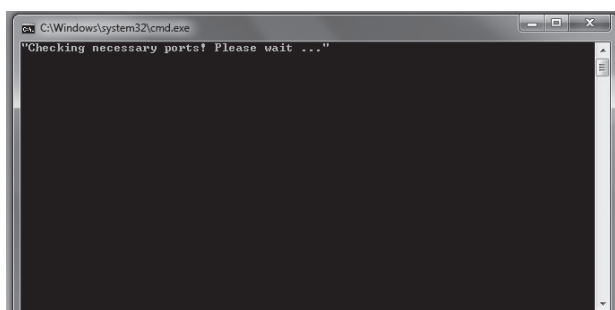


Figura 2.135. Comprobación

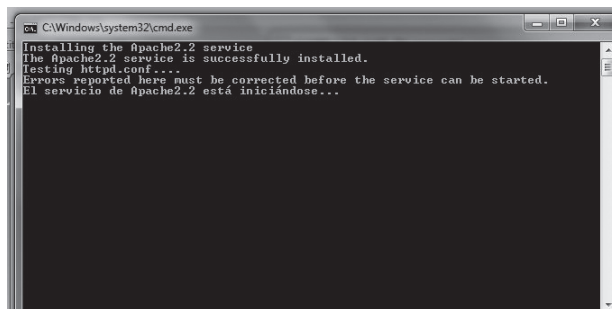


Figura 2.136. Comprobación

Para comprobar que todo esta correcto tenemos un acceso directo en el escritorio (si seleccionamos esa opción en la instalación) y un icono en la barra de tareas con el símbolo de XAMPP. Podemos pulsar sobre ese icono y comprobaremos si Apache y MySQL están funcionando, o si lo deseamos, para alguno de los servicios que están funcionando.

Si no hemos marcado la opción **Instalar** como servicios tanto Apache como MySQL y queremos utilizarlos, deberemos arrancar dichos programas manualmente desde aquí.

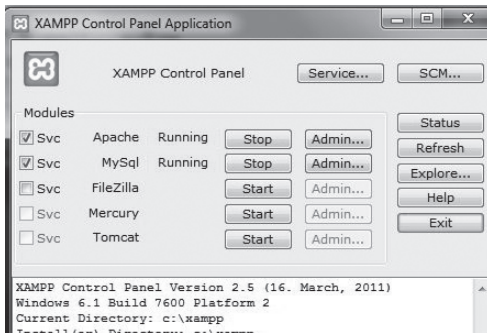


Figura 2.137. XAMPP Contro Panel



Figura 2.138. Icono de XAMPP

ACTIVIDADES 2.7



➤ Descargue, instale y configure en su ordenador o en una máquina virtual con Windows XAMPP.

2.6.2 INSTALACIÓN DE XAMPP EN UBUNTU

Lo primero que tenemos que hacer es bajar el último paquete XAMPP desde:

<http://www.apachefriends.org/en/xampp-linux.html>.

Dentro de esa página nos descargamos el fichero con la ultima versión para Linux.

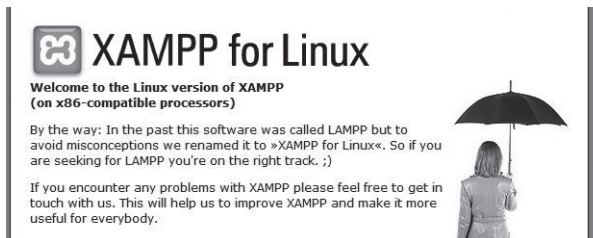


Figura 2.139. XAMPP for Linux

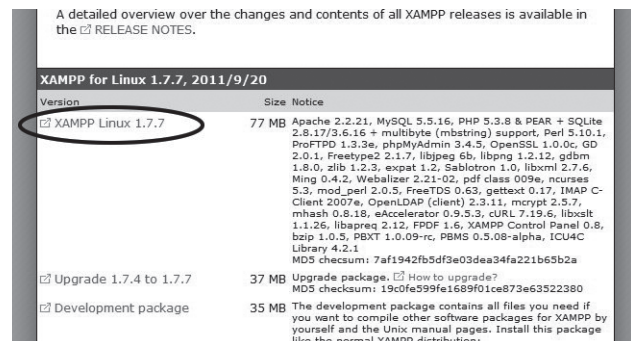


Figura 2.140. Descarga de XAMPP

Una vez descargado el fichero ejecutamos desde consola (como root):

```
sudo tar xvfz xampp-linux-X.X.X.tar.gz -C /opt
```

Eso es todo. XAMPP se instala por defecto en `/opt/lampp`. Para arrancarlo ejecutamos:

```
/opt/lampp/lampp start
```

Aparecen los mensajes de texto que indican que el servidor web y la base de datos se han iniciado:

```
da
hine:~$ sudo /opt/lampp/lampp start
```

Figura 2.141. *lampp start*

```
[sudo] password for administrador:
Starting XAMPP for Linux 1.7.7...
XAMPP: Starting Apache with SSL (and PHP5)...
XAMPP: Starting MySQL...
XAMPP: Starting ProFTPD...
XAMPP for Linux started.
administrador@administrador-virtual-machine:~$
```

Figura 2.142. *lampp start*

Podemos comprobar que se ha instalado correctamente abriendo una sesión con el navegador web y escribiendo como URL: `http://localhost`

ACTIVIDADES 2.8



➤ Descargue, instale y configure en su ordenador o en una máquina virtual con Ubuntu LAMP.

Si queremos consultar documentación existente sobre Apache, PHP, MySQL y otros temas, podemos hacerlo en: `http://localhost`. Nos vamos a **Documentación** y tendremos una serie de enlaces web para ampliar nuestros conocimientos, resolver dudas y aprender a configurar mejor y más seguro nuestro servidor.

1.7.7

[PHP: 5.3.8]

Bienvenido

Estado

chequeo de seguridad

Documentación

Componentes

PHP

phpinfo()

Administración de CD

Bioritmo

Instant Art

Agenda de telefonos

Perl

perlinfo()

Libro de invitados

Documentación Online

XAMPP combina muchos paquetes diferentes en un sólo paquete. Aquí una referencia de los paquetes más importantes de XAMPP.

- [Apache 2 Dokumentation \(en Inglés\)](#)
- [PHP Referenz-Dokumentation \(en Inglés\)](#)
- [Perl Dokumentation \(en Inglés\)](#)
- [MySQL Dokumentation \(en Inglés\)](#)
- [ADODB \(en Inglés\)](#)
- [eAccelerator \(en Inglés\)](#)
- [FPDF Class Dokumentation](#)

Y aquí una pequeña selección de las instrucciones y de las páginas con:

- [Documentación Apache Friends](#)
- [PHP para tí by David Gowans](#)
- [HTML - An Interactive Tutorial For Beginners by Dave Kristula](#)
- [Introducción a CGI by Nik Silver](#)

Mucha diversión y éxito con la lectura! :)

Figura 2.143. *Documentación*



RESUMEN DEL CAPÍTULO

En este segundo capítulo hemos tratado los requerimientos que debe tener un servidor web, tanto a nivel de hardware como a nivel de software.

También se ha explicado cómo instalar los sistemas operativos más utilizados en la actualidad en los ordenadores.

Así mismo, se ha detallado cómo hemos instalado y configurado el servidor web Apache, el preprocesados PHP y el sistema gestor de bases de datos MySQL en nuestro servidor, ya sea de forma manual o mediante XAMPP y LAMPP.

Se ha presentado una de las herramientas más importantes para la gestión de bases de datos MySQL, como es PHPMyAdmin. Hemos visto también cómo instalarla para los distintos sistemas operativos.

Para finalizar, y si queremos ampliar y mejorar nuestros conocimientos sobre Apache, PHP y MySQL, qué menos que consultar la extensa y extraordinaria documentación existente sobre dichos apartados. Como bien expresa la propia página de documentación de XAMPP (<http://localhost/xampp/manuals.php>): “¡Mucha diversión y éxito con la lectura! :)”.



EJERCICIOS PROPUESTOS

1. ¿Cuál de los tres sistemas operativos explicados le resulta más complicado de instalar? ¿Por qué?
2. Busque en Internet otros servidores web y plantee una guía de instalación similar de la de Apache, PHP y MySQL.
3. Requerimientos mínimos para instalar Ubuntu.
4. Requerimientos mínimos para instalar Windows XP.
5. Requerimientos mínimos para instalar Windows 7.
6. Según las características del hardware de su máquina, ¿qué sistema operativo instalaría?
7. Localice e instale otros programas *todo en uno* tipo XAMPP.



TEST DE CONOCIMIENTOS



- 1** Para instalar Windows XP vamos a necesitar:
- a)** El CD de instalación.
 - b)** Una unidad de CD/DVD en nuestro ordenador.
 - c)** Un disco duro.
 - d)** Todas son ciertas.
 - e)** Todas son falsas.

- 2** Los requisitos mínimos para instalar Windows 7 son:
- a)** Procesador de 2 GHz.
 - b)** 2 GB de disco duro.
 - c)** 16 ó 20 GB de memoria RAM.
 - d)** Todas son ciertas.
 - e)** Todas son falsas.

- 3** Los requisitos mínimos para instalar Ubuntu 11.10 son:
- a)** Procesador de 2 GHz.
 - b)** 2 GB de disco duro.
 - c)** 16 ó 20 GB de memoria RAM.
 - d)** Todas son ciertas.
 - e)** Todas son falsas.

- 4** Uno de los servidores web más conocidos y utilizados del mercado es:
- a)** MySQL.
 - b)** PHP.
 - c)** Apache.
 - d)** PHPMyAdmin.
 - e)** Todas son ciertas.
 - f)** Todas son falsas.

- 5** Para instalar MySQL en Windows es necesario tener instalado:
- a)** PHP.
 - b)** Oracle.
 - c)** Net Framework.
 - d)** Zend Framework.
 - e)** Todas son ciertas.
 - f)** Todas son falsas.

3

Gestores de contenidos

OBJETIVOS DEL CAPÍTULO

- ✓ Conocer qué son los CMS.
- ✓ Conocer el funcionamiento y la estructura de los CMS.
- ✓ Distinguir distintos tipos de CMS.

3.1 DEFINICIÓN DE GESTOR DE CONTENIDOS O CMS

Un Sistema de gestión de contenidos, *Content Management System* (CMS), es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente, por parte de los participantes.

Es un software instalado en el servidor que nos permite crear una Web con publicación descentralizada, sin necesidad de conocer el lenguaje HTML para publicar todo tipo de textos, archivos, etc. La publicación se realiza desde el mismo navegador.

Una herramienta CMS, generalmente, contendrá una interfaz basada en formularios, a los que habitualmente se accede con el navegador, donde se pueden dar de alta los contenidos fácilmente. Esos contenidos luego aparecerán en la página en los lugares donde se ha indicado al darlos de alta. Por lo tanto, un CMS estará compuesto de dos partes, un *back* y un *front*, siendo el *back* la parte donde los administradores publican las informaciones y el *front* la parte donde los visitantes visualizan las mismas.

3.1.1 HISTORIA DE LOS CMS

Los primeros sistemas de administración de contenidos fueron desarrollados por organizaciones que publicaban una gran cantidad de contenido en Internet y necesitaban de continuas actualizaciones; como revistas en línea, periódicos y publicaciones corporativas.

En 1995, el sitio de noticias tecnológicas CNET sacó su sistema de administración de documentos y publicación y creó una compañía llamada Vignette, pionera de los sistemas de administración de contenido comerciales.

La evolución de Internet hacia portales con más contenido y la alta participación de los usuarios directamente, a través de *blogs* y redes sociales, han convertido a los gestores de contenidos en una herramienta esencial en Internet, tanto para empresas e instituciones como para las personas.

Las wikis y los sistemas *groupware* también son considerados CMS.

Hoy en día existen sistemas desarrollados en software libre y no libre. En ambos casos es necesaria una implantación para adaptar el gestor de contenidos al esquema gráfico y funcionalidades deseadas. Para ciertos gestores existen muchas plantillas disponibles que permiten una sencilla implantación por parte de un usuario sin conocimientos de programación.

3.1.2 FUNCIONAMIENTO DE UN CMS

Un sistema de administración de contenido siempre funciona en el servidor web en el que esté alojado el portal. El acceso al gestor se realiza generalmente a través del navegador web y se puede requerir el uso de FTP para subir contenido.

Cuando un usuario accede a una URL, se ejecuta en el servidor esa llamada, se selecciona el esquema gráfico y se introducen los datos que correspondan de la base de datos. La página se genera dinámicamente para ese usuario. El código HTML final se genera en esa llamada. Normalmente, se predefinen en el gestor varios formatos de presentación de contenido para darle la flexibilidad a la hora de crear nuevos apartados e informaciones.

3.1.3 ¿PARA QUIÉN VAN DESTINADOS LOS CMS?

Los Sistemas Gestores de Contenidos, entre otros, podrían ir destinados a:

- ✓ Empresas que tengan vocación de comunicarse con sus potenciales clientes a través de la web y a aquellas que necesiten proveer de información actualizada a sus clientes, como las pymes (catálogos, folletos, artículos...).
- ✓ Corporaciones (noticias, productos, comunicados financieros...).
- ✓ Páginas personales, redes sociales, etc. (esto está muy en boga últimamente con las aplicaciones como Blogger, Wordpress, etc., donde cualquier persona puede convertirse en un medio de comunicación digital).

3.2 CONCEPTOS GENERALES DE UN CMS

Es entendido como un sistema de soporte a la gestión de contenidos, ya que, en realidad, son las estrategias de comunicación las que realmente llevan a gestionar contenidos y publicidad de forma efectiva. Los sistemas informáticos pueden, a lo sumo, proporcionar las herramientas necesarias para la publicación en línea, o bien incluir servicios de soporte a la toma de decisiones, por lo que a la gestión de contenidos se refiere.

El gestor de contenidos se aplica generalmente para referirse a sistemas de publicación, pudiendo subestimarse las funcionalidades de soporte y mantenimiento, en detrimento de las funcionalidades relacionadas con la optimización de los tiempos de publicación. La correcta implantación del sistema, con arreglo a las necesidades del cliente es obligada y es necesario entender el proyecto de un portal web en el seno de un proyecto de comunicación estructurado y bien planteado.

La elección de la plataforma correcta será vital para alcanzar los objetivos del cliente, ya que existen particularidades diferenciales tanto en su adaptabilidad a esquemas gráficos como a la posible integrabilidad de funcionalidades y extensiones adicionales.

El posicionamiento en buscadores está relacionado con el volumen de contenidos de un portal y con la forma en la que éste se presenta. Es importante tener eso en cuenta para la estructura del portal y así garantizar un correcto posicionamiento orgánico.

3.3 ESTRUCTURA DE UN CMS

Las diferentes funcionalidades ofrecidas por un CMS pueden ser separadas en diferentes categorías.

- **Front-end y Back-end.** Un CMS consiste en un *Front-end*, que es el sitio web (*website*), el cual los visitantes y los usuarios registrados pueden ver. Y un *Back-end*, que contiene la parte de administración, localizado en una diferente URL al sitio web, es donde se realizan las tareas de configuración, mantenimiento, limpieza, creación de estadísticas, etc.

- **Configuración.** Las decisiones de configuración que se aplican al sitio web son especificados en *Configuration settings*. Esto incluye el título del sitio web, las palabras para los motores de búsqueda, opciones que permiten o prohíben darse de alta como usuario en el sitio web y muchas otras funciones.
- **Derechos de acceso.** En un CMS, los nombres de usuario son asignados a personas que están involucradas en el desarrollo del sitio web y su administración y poseen diferentes derechos de acceso. El rango de derechos va desde un usuario simplemente registrado, como autor o editor, hasta la categoría de administrador, que tiene total control sobre el dominio. Según los derechos asignados, el sitio web muestra diferentes contenidos,
- **Contenido.** Un CMS permite cualquier tipo de contenido. Texto simple, imágenes, vínculos, música y archivos multimedia en general o una combinación de todos ellos. Para mantener una buena presentación de los contenidos, son incluidos en estructuras jerarquizadas, siendo clasificados según secciones y categorías que, a su vez, deben ser administradas.
- **Plantillas.** Una plantilla define los colores, las fuentes y su tamaño, imágenes de fondo, espaciado y distribución de la página, es decir, todo lo que tiene que ver con la apariencia del sitio web.
- **Extensiones.** Todo sistema debe poder ser ampliable para poder crecer en función de los requerimientos que aparezcan. Las funcionalidades que pertenecen a determinado contexto son cubiertas, por lo que conocemos como componente. Por ejemplo, componentes típicos son una tienda *on line*, un gestor de usuarios, un gestor de listas de correo, un foro, una galería de imágenes, un gestor de descargas, etc. Los módulos que se necesitan integrar con los componentes son utilizados para insertar contenidos en la parte deseada dentro de la plantilla. Son ejemplos de módulos: los que permiten ver a los usuarios conectados en el momento, los que permiten ver las estadísticas del sitio, los que muestran los artículos de contenido más recientes, etc. Son consideradas extensiones también las plantillas, los paquetes de idiomas y, también, los llamados *plugins*, cuya funcionalidad principal, pero no la única, es interceptar los contenidos y realizar alguna modificación en ellos antes de ser mostrados en el navegador.

3.3.1 RUTINA DE TRABAJO

Si hay varias personas trabajando con el CMS, una buena rutina de trabajo facilitará la gestión de los contenidos. Por ejemplo, si un editor revisa una lista de noticias enviadas al sitio para ser corregidas y después de revisarlos indica que las noticias ya han sido corregidas. El jefe de redacción podrá directamente decidir si la noticia se publica en el *FrontPage* (página principal del sitio web) o no.

3.3.2 GESTIÓN DE USUARIOS

Dependiendo de la plataforma escogida se podrán escoger diferentes niveles de acceso para los usuarios; yendo desde el administrador del portal hasta el usuario, o creador de contenido, sin permiso de edición. Dependiendo de la aplicación podrá haber varios permisos intermedios que permitan la edición del contenido, la supervisión y reedición del contenido de otros usuarios, etc.

El sistema de gestión de contenidos controla y ayuda a manejar cada paso de este proceso, incluyendo las labores técnicas de publicar los documentos a uno o más sitios. En muchos sitios con estos sistemas una sola persona hace el papel de creador y editor como, por ejemplo, en los *blogs* personales.

3.4 TIPOS DE CMS

Existen distintos tipos de CMS. De acuerdo con la misma definición de CMS queda claro que cualquier sistema que implemente una Gestión de Contenidos puede considerarse como un CMS. Con ello, podemos distinguir en esta etapa las categorías más representativas:

- **Gestión de Portales.** Estos CMS sirven para gestionar todo el contenido de un sitio web ofreciendo funcionalidad diversa. Uno de los principales es Joomla (<http://www.joomla.org>), que ha ganado amplia popularidad convirtiéndose en paradigmático en esta categoría por sus facilidades para gestionar un sitio web

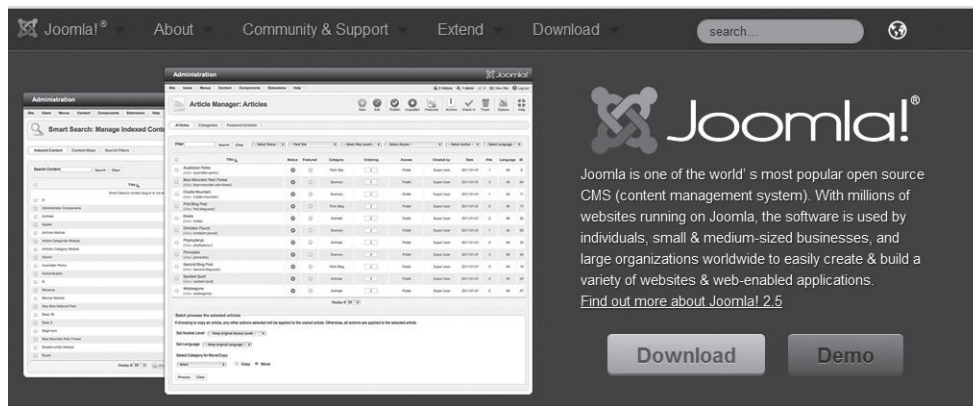


Figura 3.1. Gestión de Portales

- **Blogs.** Estos permiten la publicación de noticias en orden cronológico, permitiendo comentarios. Dos de los más populares ya se conocieron en el módulo anterior: WordPress y Blogger. En los últimos tiempos se han hecho populares diferentes artefactos (*gadgets*), en particular por Google, que permiten incorporar en los *blogs* diferentes utilidades, como diccionarios, citas célebres, seguimiento de lectores, estado del tiempo, noticias, etc.



Figura 3.2. Blogs

- **Gestores de Foros.** Estos permiten la discusión en línea entre usuarios: phpBB, SMF, MyBB.

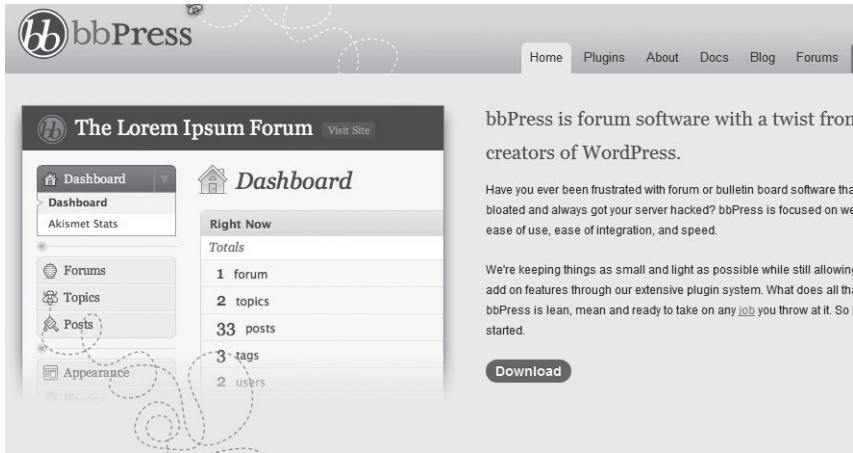


Figura 3.3. Foros

- **Wikis.** La wiki es un tipo especial de web que permite la creación colaborativa de contenidos. El ejemplo más destacado es la Wikipedia. Otros gestores de Wiki son: MediaWiki, TikiWiki.



Figura 3.4. Wikis

- **Gestores de Comercio Electrónico.** Estos permiten generar sitios web específicos para comercio electrónico. Por ejemplo: *os-commerce* (licenciaGPL), *Dynamicweb e-commerce* (comercial).

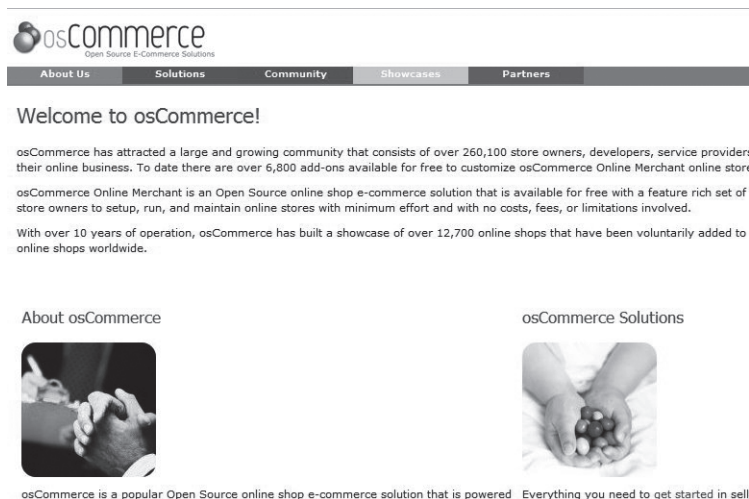


Figura 3.5. Gestores comercio

- **Galerías.** Estos permiten administrar y generar automáticamente un portal o sitio web que muestre contenido audiovisual, normalmente imágenes. Por ejemplo: Gallery (<http://gallery.menalto.com>) y Dragonfly CMS.

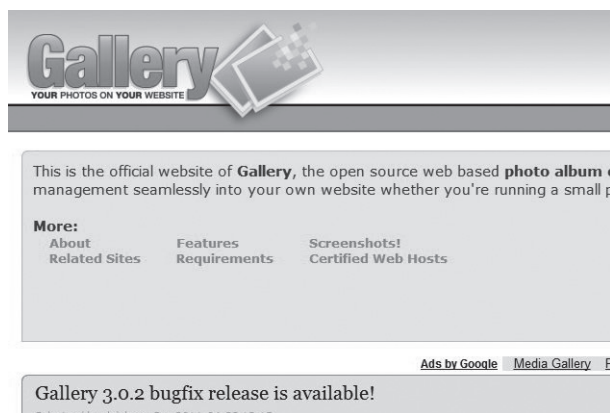


Figura 3.6. Galerías

- **Gestores de e-learning.** Estos son los denominados LMS (*Learning Manadgment Systems*). Las particularidades del proceso de enseñanza-aprendizaje han hecho que se desarrollen diferentes tipos de plataformas o LMS. El gestor de LMS más importante y conocido es Moodle (<http://moodle.org/>), que se ha convertido en uno de los más utilizados a nivel mundial.

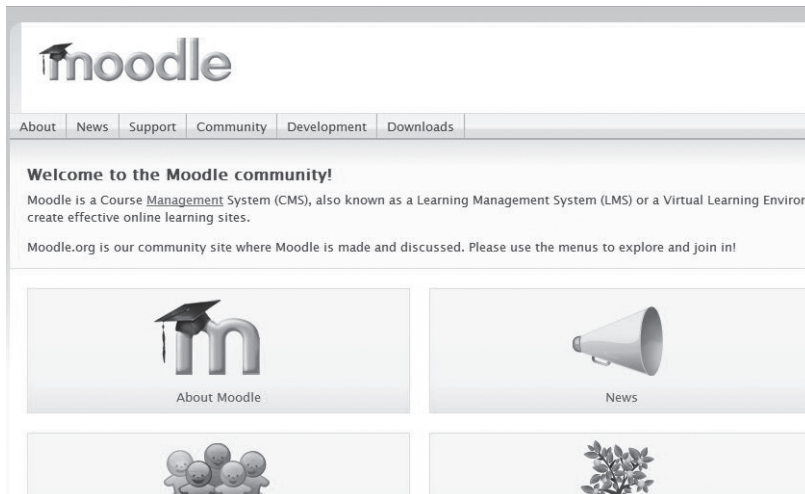


Figura 3.7. Gestores e-learning

- **Otros.** Existen también plataformas especialmente diseñadas para temas concretos, teniendo en cuenta las necesidades de las publicaciones digitales, tales como periódicos, revistas, etc. Por ejemplo: ePrints, Thihkdot CMS.

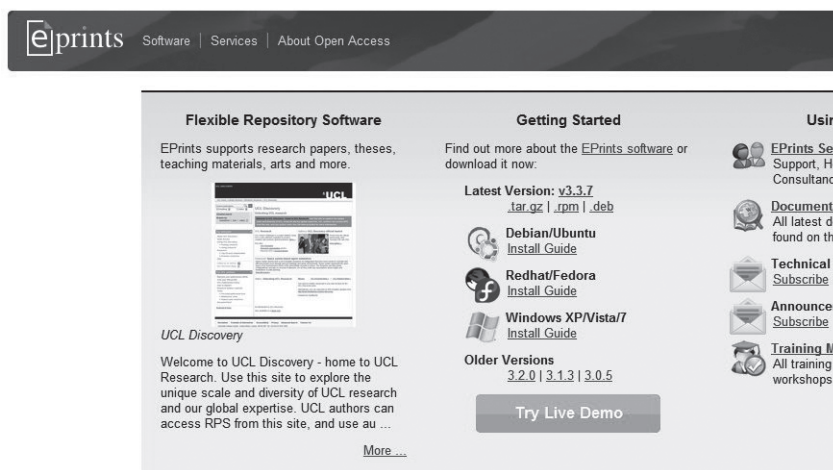


Figura 3.8. Otros

3.5 LIMITACIONES DE LOS CMS

Todo depende de cuál nos instalemos y dónde. **Cuál** porque algunos no están traducidos a nuestro idioma y porque debemos elegir uno adecuado a nuestro proyecto, que no se nos quede grande puesto que nos complicaría la gestión, ni corto, porque no nos permitiría desarrollar la web según nuestras expectativas. **Dónde** nos lo marca nuestro servidor. Evidentemente, los gratuitos ofrecen pocas posibilidades, pero por precios razonables se consiguen alojamientos con los requisitos necesarios.

3.6 REQUERIMIENTOS PARA INSTALAR UN CMS

La inmensa mayoría están escritos en PHP, por lo que nuestro servidor debe disponer de esta herramienta. Casi todos los servidores funcionan con sistemas Linux y, por tanto, esta opción viene por defecto.

Además, necesitamos que nuestro servidor nos proporcione bases de datos, generalmente MySQL, al menos una, pues es ahí donde se van a escribir los textos que incluyamos. Al igual que en el caso del PHP, los servidores basados en Linux también tienen el software MySQL por defecto. Así mismo, necesitamos acceso FTP a nuestro servidor.



RESUMEN DEL CAPÍTULO

En este capítulo hemos introducido un concepto que nos acompañará en lo sucesivo (y puede que el resto de nuestra vida): los Sistemas de Gestión de Contenidos, o CMS.

A lo largo del capítulo hemos visto un poco de historia, la estructura y los distintos tipos de CMS que existen. A su vez, se han presentado tanto las limitaciones como los requerimientos para poder utilizar estos CMS.



EJERCICIOS PROPUESTOS

- 1. Defina con sus palabras qué es un CMS.
- 2. ¿Por qué son tan importantes hoy en día los CMS?
- 3. Explique el funcionamiento de un CMS.
- 4. Tipos de CMS.
- 5. Enumere las distintas partes de la estructura de un CMS.
- 6. De todos los CMS que se han tratado en el tema, haga una lista de los que conoce y si los ha utilizado alguna vez.



TEST DE CONOCIMIENTOS

- 1** Un CMS es un sistema que:
- a) Nos permite crear webs de manera muy sencilla.
 - b) Nos permite crear webs sin tener elevados conocimientos informáticos.
 - c) Puede utilizarlo todo el mundo.
 - d) Generalmente es de tipo gratuito.
 - e) Todas son ciertas.
 - f) Todas son falsas.

- 2** Un tipo de CMS son:
- a) Gestión de foros.
 - b) Gestión de redes.
 - c) Gestión de recursos.
 - d) Todas son ciertas.
 - e) Todas son falsas.

- 3** Los gestores de *e-learning*, como Moodle, también se conocen como:
- a) CMD.
 - b) LMS.

- c) DDS.
- d) CTL.
- e) DDR.
- f) Todas son falsas.

- 4** Para instalar un CMS necesitamos:
- a) Un sistema multiprocesador.
 - b) Un servidor web sin soporte PHP.
 - c) Un servidor web con soporte PHP.
 - d) Nada en especial.
 - e) Todas son falsas.

- 5** El primer CMS apareció en el año:
- a) 1990.
 - b) 1992.
 - c) 1995.
 - d) 1998.
 - e) 2011.

4

Blogger

OBJETIVOS DEL CAPÍTULO

- ✓ Conocer qué es y qué partes tiene un *blog*.
- ✓ Conocer qué es Blogger y su historia.
- ✓ Crear una cuenta Google.
- ✓ Crear un *blog*, con Blogger, dotándolo de contenido.

4.1 ¿QUÉ ES UN BLOG?

El término *log*, en inglés, se utiliza para hacer referencia a un archivo que contiene el registro de actividad de un servidor. Por extensión, en el ámbito de Internet, se utilizó el término *weblog*, para representar el diario de navegación por la Web de un internauta que, en la actualidad y de forma abreviada, se simplifica en *blog*.

También, en español, se utiliza la expresión *bitácora* (en su analogía a cuaderno de bitácora), haciendo referencia a estos elementos de navegación por mar.

La Wikipedia también escribe sobre qué es un *blog*: “Un *blog*, o en español también una bitácora, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente. El término *blog* proviene de las palabras *web* y *log* (*log*, en inglés = diario). El término bitácora, en referencia a los antiguos cuadernos de bitácora de los barcos, se utiliza preferentemente cuando el autor escribe sobre su vida propia como si fuese un diario, pero publicado en Internet en línea”.

“Habitualmente, en cada artículo de un *blog*, los lectores pueden escribir sus comentarios y el autor darles respuesta, de forma que es posible establecer un diálogo. No obstante es necesario precisar que ésta es una opción que depende de la decisión que tome al respecto el autor del *blog*, pues las herramientas permiten diseñar *blogs* en los cuales no todos los internautas (o incluso ninguno) puedan participar. El uso o tema de cada *blog* es particular, los hay de tipo personal, periodístico, empresarial o corporativo, tecnológico, educativo (*edublogs*), políticos, etc.”.

Según Antonio Fumero, en Web 2.0, Fundación Orange, el fenómeno de los *blogs* ha supuesto una auténtica revolución: “El impacto de la aparición del *blog* (considerado como herramienta personal para la comunicación y la expresión individual en la Red) en la evolución histórica de la Web puede asimilarse con la popularización de los ordenadores personales a partir del lanzamiento del PC por parte de IBM (un hito que conmemoró su primer cuarto de siglo en agosto de 2006) que, no siendo la primera concepción comercial de aquellas máquinas informáticas pioneras de una era postindustrial, sí dio con la fórmula para difundir entre el usuario medio una herramienta de excepcional potencialidad, como la historia ha demostrado posteriormente”.

4.2 ESTRUCTURA DE UN BLOG

Básicamente, en un *blog* podemos encontrar los siguientes elementos:

4.2.1 LA CABECERA

Es un texto o una imagen que identifican al *blog*. Forma parte del diseño que se ha elegido para el mismo.

4.2.2 EL POST O ENTRADA

Cada *post* o entrada también suele estar acompañado de la categoría o etiqueta en la que el autor lo ha incluido. Es una forma de organizar la información que se publica. Un *post* puede pertenecer a varias categorías o etiquetas. Éstas servirán para realizar búsquedas en el *blog* por estos mismos términos. Es el elemento fundamental. En él el autor o autores escriben sus opiniones, artículos, etc., sobre la temática bajo la que se publica el *blog*.

Suele ir acompañado, cada vez con más frecuencia, de elementos multimedia, como fotos, vídeos o reproductores de sonido, que complementan la publicación, cuando no son el principal motivo



Figura 4.1. La Cabecera

Bajo la sombra del Cine Negro.



"Los últimos rayos del crepúsculo se esconden entre los edificios provocando un tortuoso baile de tenues sombras que se mueven al compás de la ciudad. Una silueta, ataviada con una gabardina y un sombrero

de ala ancha, se encuentra subiendo las escaleras del edificio de su vivienda. Al llegar descubre la llave de su puerta, escondida en el bolsillo de la gabardina, sin poder evitar escuchar a los vecinos discutir, otra vez, a grito pelado. Seguramente el marido habría vuelto a llegar borracho a casa y estaría pagando sus frustraciones y rabias con su pobre mujer. (más...)

Figura 4.2. Post o Entrada

4.2.3 LOS COMENTARIOS

Cada *post* o entrada puede recibir comentarios de los usuarios que la leen, creándose una verdadera conversación en torno al tema publicado. Así los lectores también son protagonistas en la publicación a través de sus comentarios.

El autor del *blog* tiene la posibilidad de permitir o no la entrada de comentarios mediante la configuración que ofrece la herramienta con la que se mantiene el *blog*. Recibir comentarios en un *blog* es una de las mayores satisfacciones de los *bloggers*.

4.2.4 EL BLOGROLL

Es una colección de enlaces de *blogs* para encontrar más *blogs*. Los autores de *blogs* pueden definir diferentes criterios para incluir otros *blogs* en sus *blogroll*. Habitualmente, el listado se compone de webs que los propios autores visitan con asiduidad o a veces simplemente de *blogs* de amigos o cercanos.

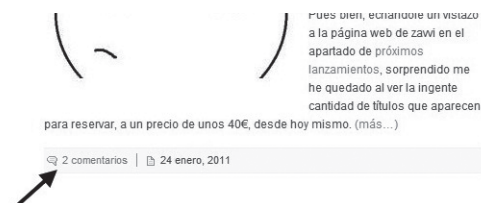


Figura 4.3. Comentarios

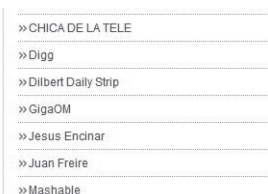


Figura 4.4. Blogroll

4.2.5 FECHA/HORA

Cada *post* o entrada suele estar acompañado de su fecha y hora de publicación.

4.2.6 CATEGORÍAS/ETIQUETAS

Cada *post* o entrada también suele estar acompañado de la categoría o etiqueta en la que el autor lo ha incluido. Es una forma de organizar la información que se publica. Un *post* puede pertenecer a varias categorías o etiquetas. Estas servirán para realizar búsquedas en el *blog* por estos mismos términos.

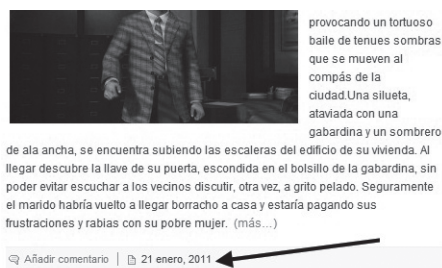


Figura 4.5. Fecha / Hora



Figura 4.6. Etiquetas

4.2.7 ENLACE PERMANENTE O PERMALINK

Es una dirección URL fija que identifica al *post* individualmente, de manera que con ella es posible acceder a esa entrada sin presentar o visualizar otras anteriores o posteriores. Es útil para hacer referencia a un artículo concreto publicado en un *blog*, desde otros *blogs* o páginas web.

4.2.8 SINDICACIÓN

Todos los *blogs* ofrecen sindicación de contenidos o RSS, es decir, publican mediante los canales o *feeds*, versiones de sus publicaciones para que otros usuarios, mediante los agregadores, para que puedan estar al corriente de las novedades que se produzcan en ellos.

4.2.9 WIDGETS O GADGETS

Son elementos que los autores de los *blogs* insertan en alguna columna de los mismos y que ofrecen algunas funcionalidades más. Por ejemplo, las últimas fotos subidas a Flickr, un reloj, los últimos comentarios publicados en Twitter, etc.

4.3 BLOGGER

Blogger es un servicio creado por Pyra Labs para crear y publicar una bitácora en línea. El usuario no tiene que escribir ningún código o instalar programas de servidor o de *scripting*. Blogger acepta para el alojamiento de las bitácoras su propio servidor (*Blogspot*) o el servidor que el usuario especifique (FTP o SFTP).

4.3.1 HISTORIA DE BLOGGER

Lanzado en agosto de 1999, es una de las primeras herramientas de publicación de bitácora en línea y es acreditado por haber ayudado a popularizar el uso de formularios. Más específicamente, en vez de escribir a mano el HTML y subir las nuevas publicaciones, el usuario puede publicar en su bitácora en línea llenando un formulario en el sitio web de Blogger. Esto puede ser realizado por cualquier navegador y los resultados son inmediatos.

En el 2003, Pyra Labs fue adquirido por la empresa Google; por ende, también Blogger. Google consiguió los recursos que Pyra requería. Más adelante, las “características premium”, que eran de pago, fueron habilitadas para el público gracias a la ayuda de Google.

En el 2004, Google compró Picasa y su utilidad de intercambio de fotografías Hello. Esto permitió a los usuarios de Blogger poner fotografías en sus bitácoras. Así el *photoblogging* (o la posibilidad de publicar fotografías en las bitácoras) se hacía realidad en Blogger con la integración de Hello. El 9 de mayo de 2004, Blogger fue relanzado añadiendo nuevas plantillas de diseño basadas en CSS, archivaje individual de publicaciones, comentarios y publicación por correo electrónico.

A finales de 2006, con el nuevo Blogger Beta, se hizo posible lo que tanta falta hacía en Blogger: el poder publicar artículos por categorías o etiquetas, (*labels*) como son llamados en Blogger, así como la posibilidad de poner bitácoras de acceso restringido o privadas para solo unos cuantos, entre otras funciones.

Inicialmente, el servicio de Blogger Beta no permitía hacer los cambios a la plantilla modificando el código HTML. Tiempo después se migraron las cuentas anteriores de Blogger al nuevo Blogger Beta sin que causara ninguna molestia a sus usuarios.

La actualización al nuevo Blogger requiere el registro de una cuenta de Google e incluye, entre otras mejoras, el servicio de etiquetado de artículos y una mejora en la interfaz de edición y publicación de artículos. La nueva versión no modifica en absoluto la apariencia de las bitácoras en línea creados en la versión antigua, salvo en algunos pequeños problemas con tildes y caracteres especiales, como la letra ñ. Una vez actualizado un *blog*, puede decidirse dar un paso más y actualizar la plantilla, con lo que se perderían las modificaciones realizadas al HTML.

4.4 CREACIÓN DE UNA CUENTA DE CORREO ELECTRÓNICO CON GMAIL

En primer lugar, nos dirigimos a la dirección: <http://www.gmail.com>. Pulsamos sobre el botón **CREAR UNA CUENTA** y rellenamos todos los datos solicitados.

Figura 4.7. www.gmail.com

Figura 4.8. Crear una cuenta

Una vez completados todos los datos debemos marcar la casilla **Acepto las Condiciones del servicio y la política de privacidad de Google**. Pulsamos sobre **Siguiente paso** y nos aparecerá una pantalla de bienvenida a nuestro correo.

Figura 4.9. Acepto las condiciones

Figura 4.10. Cuenta creada

Ya podemos disfrutar de nuestra nueva cuenta de correo electrónico, además de regalo tenemos una **Cuenta Google**, la cual no permitirá acceder a todos los contenidos ofertados por Google, como Blogger y Google Docs, por poner unos ejemplos.

ACTIVIDADES 4.1



- Cree una nueva cuenta en Gmail, haga capturas de todos los pasos que ha ido realizando para la creación de dicha cuenta.

4.6 CREACIÓN DE NUESTRO BLOG CON BLOGGER

Nos dirigimos a la página web <http://www.blogger.com>. En esta página debemos identificarnos con nuestra cuenta Google para poder acceder a los servicios de Blogger.

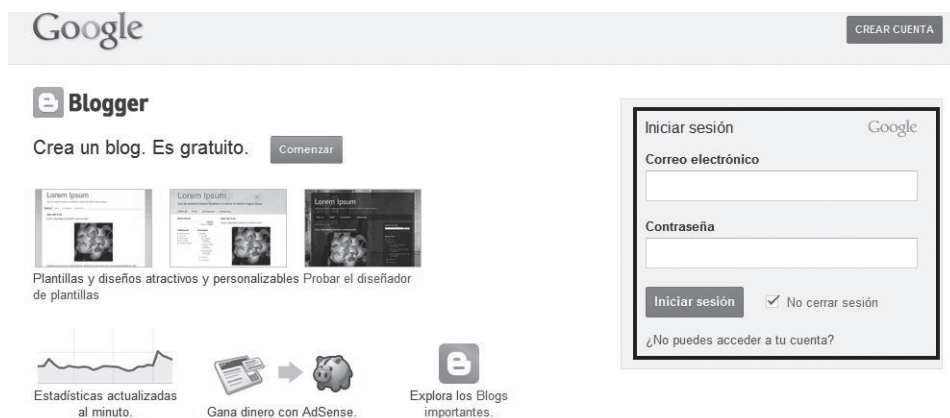


Figura 4.11. www.blogger.com

Una vez nos hemos indentificado con nuestra **cuenta de Google**, procedemos a registrarnos en **Blogger**. Para ello debemos rellenar el nombre visible. No se nos olvide **aceptar las condiciones de servicio** para poder **Continuar**.

En la siguiente pantalla debemos introducir un nombre para el **blog** (**Título del blog**) y la **dirección** que asignaremos a nuestro **blog**, que será del tipo **minombre.blogspot.com**. Podemos comprobar si la dirección elegida está disponible pulsando sobre **Comprobar la disponibilidad**.

Figura 4.12. Registrarse en Blogger

Figura 4.13. Asignar nombre al blog

Elegimos la plantilla de Inicio y clicamos sobre **CONTINUAR**. Ya tenemos nuestro *blog* creado y listo para añadir entradas o personalizar la apariencia.



Figura 4.14. Elegir una plantilla



Figura 4.15. Se ha creado su blog

ACTIVIDADES 4.2



- Cree un nuevo *blog* con Blogger, el nombre asignado para el nuevo *blog* será el nombre y apellidos del alumno/a. Haga capturas de todo el proceso.

4.6.1 PERSONALIZAR APARIENCIA DE NUESTRO BLOG

Para cambiar la plantilla, debemos ir a **Diseño** y, de ahí, nos vamos al **Diseñador de plantillas**. En este sitio podemos elegir la plantilla que más nos guste para nuestro *blog*.

Una vez hecha la personalización oportuna, pulsamos en el botón **Aplicar al blog**. Después de esto hacemos clic sobre el enlace **Volver a Blogger**.

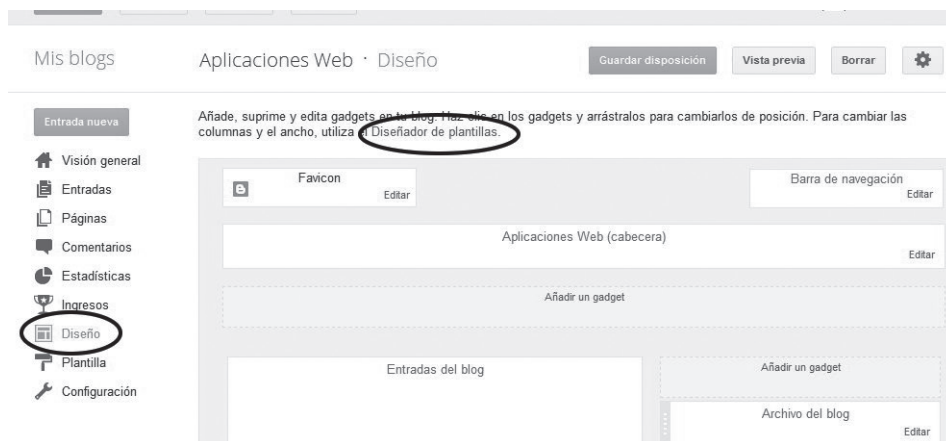


Figura 4.16. Diseñador de plantillas



Figura 4.17. Plantillas

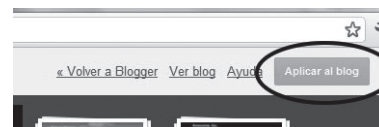


Figura 4.18. Aplicar al blog

4.6.2 CREACIÓN DE ENTRADAS EN NUESTRO BLOG

Ya podemos crear entradas en nuestro *blog*, para que el resto del mundo pueda verlas. La forma más rápida es pulsar sobre el botón **Crear entrada nueva** (como puede verse en la Figura 4.19).

Escribimos la entrada que queramos publicar en nuestro *blog*. Una vez estemos conformes con lo que queremos publicar haremos clic sobre **Publicar** y dicha entrada formará parte de nuestro nuevo *blog*.

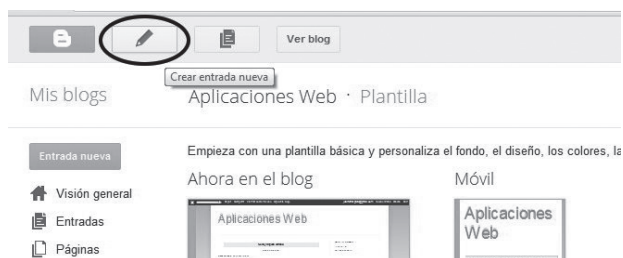


Figura 4.19. Crear entrada nueva

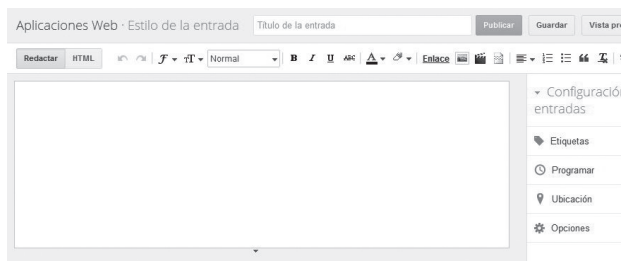


Figura 4.20. Creación de entrada

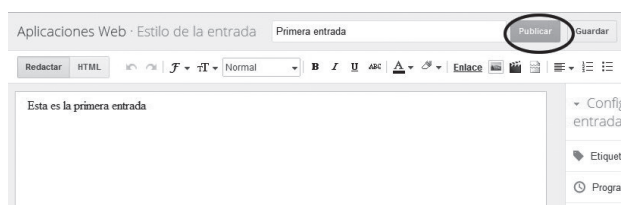


Figura 4.21. Publicar entrada



Figura 4.22. Entrada publicada

ACTIVIDADES 4.3



- Cree una entrada nueva en su *blog* y haga capturas de todo el proceso seguido.

PRÁCTICA 4.1



➤ Inserción de una entrada con texto con formato e imágenes. Ya hemos visto cómo publicar una entrada de texto en nuestro *blog*. En este caso práctico vamos a ver que además de texto plano, podemos darle formato a ese texto (negrita, cursiva, subrayada, cambiar el color, etc.) e insertar imágenes en las entradas de nuestro *blog*.

Partimos de **Crear entrada nueva**, escribimos el título del *post* y el contenido de texto que tendrá la entrada. En esta entrada vamos a introducir un texto al que vamos a aplicarse diversos formatos.



Figura 4.23. Crear una nueva entrada

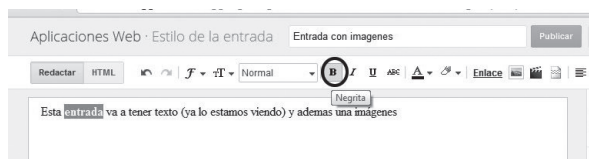


Figura 4.24. Poner texto en negrita

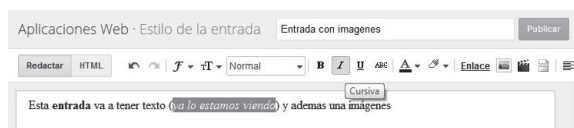


Figura 4.25. Poner texto en cursiva

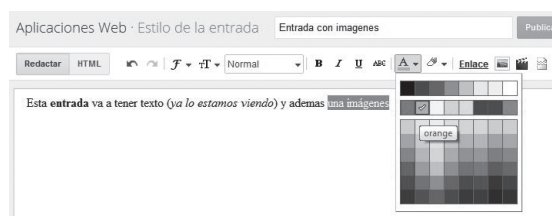


Figura 4.26. Cambiar color texto

Una vez tenemos el texto formateado, vamos a insertar una imagen. Para ello pulsamos sobre el botón **Insertar imagen**, que tiene aspecto de un paisaje (ver la Figura 4.27). Nos aparece la opción de **Seleccionar un archivo**, podemos subirlo desde nuestro equipo, desde el *blog*, desde Picasa, desde tu teléfono o desde otra web.

En nuestro caso vamos a elegir un archivo que tenemos guardado en nuestro ordenador. Pulsamos sobre **Elegir archivos** y seleccionamos el fichero a subir (Figura 4.29). Una vez seleccionada la imagen, será subida al servidor de Google.

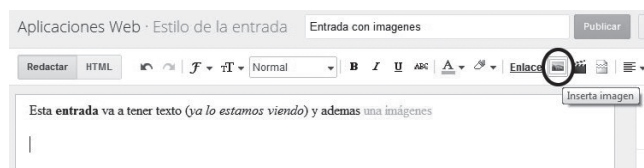


Figura 4.27. Insertar imagen

Seleccionar un archivo



Figura 4.28. Seleccionar un archivo

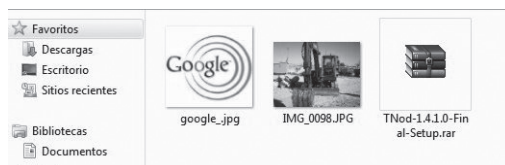


Figura 4.29. Elegimos el archivo a subir

Seleccionar un archivo



Figura 4.30. Carga del archivo

Ya tenemos la imagen en nuestra entrada, tan solo debemos pulsar sobre **Publicar** una vez hayamos finalizado la introducción de información (podemos mezclar texto e imágenes al gusto) y tendremos disponible nuestra nueva entrada.

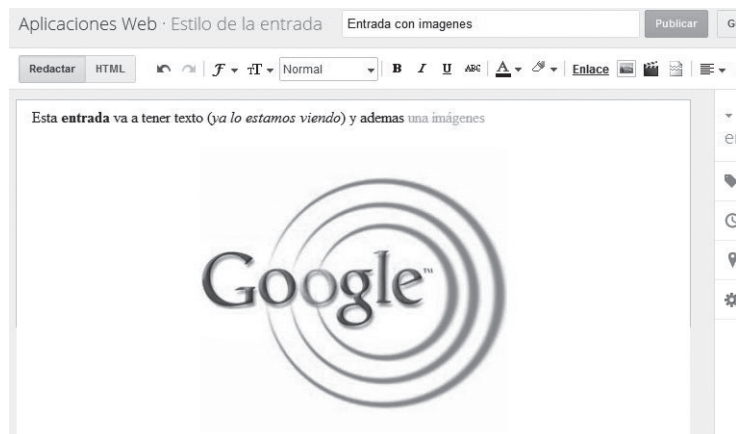


Figura 4.31. Imagen en nuestra entrada

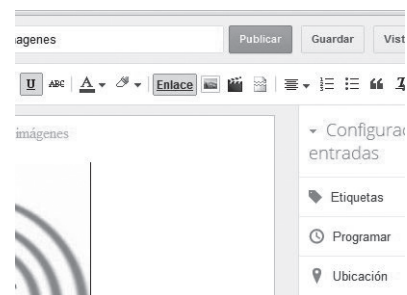


Figura 4.32. Publicar entrada

ACTIVIDADES 4.4



- Cree una nueva entrada en su *blog* incorporando una imagen con texto con formato, como lo visto en la Práctica 4.1.

PRÁCTICA 4.2



➤ Inserción de una entrada con un vídeo alojado en YouTube.

Para insertar vídeos, partimos, al igual que en caso anterior, desde **Crear entrada nueva**. Pulsamos sobre el icono **Insertar un vídeo**. Es un icono que tiene forma de claqueta (ver Figura 4.33). El comportamiento es similar a insertar imagen, nos aparecerá **Seleccionar un archivo**, y a su vez varias posibilidades para subir estos vídeos. La primera opción, **Subir**, nos permitira subir vídeos desde nuestro equipo, también podemos subirlos **desde Youtube**, o incluso desde el teléfono.

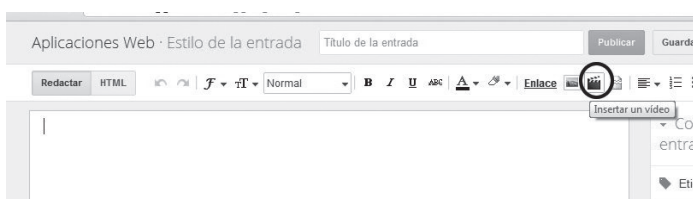


Figura 4.33. Insertar un video



Figura 4.34. Seleccionar un archivo

En nuestro caso elegimos subirlo desde Youtube. En este caso nos aparecerá un cuadro de texto que nos permite buscar el vídeo deseado en Youtube.



Figura 4.35. Archivo de Youtube

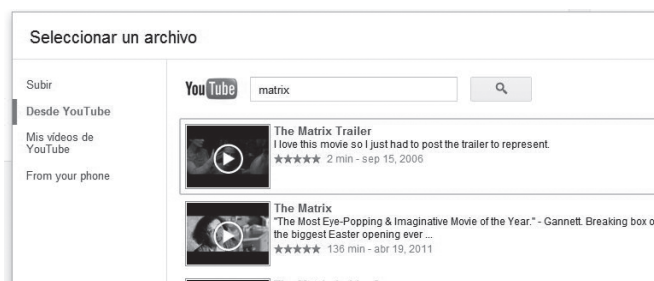


Figura 4.36. Búsqueda de vídeo

Una vez hemos encontrado el vídeo que estábamos buscando, pulsamos en **Seleccionar** y el vídeo pasará a formar parte de la entrada de nuestro *blog*, siempre y cuando pulsemos sobre **Publicar**. Podemos añadir a nuestra entrada tantos vídeos como deseemos y mezclarlos con texto, fotos y demás.

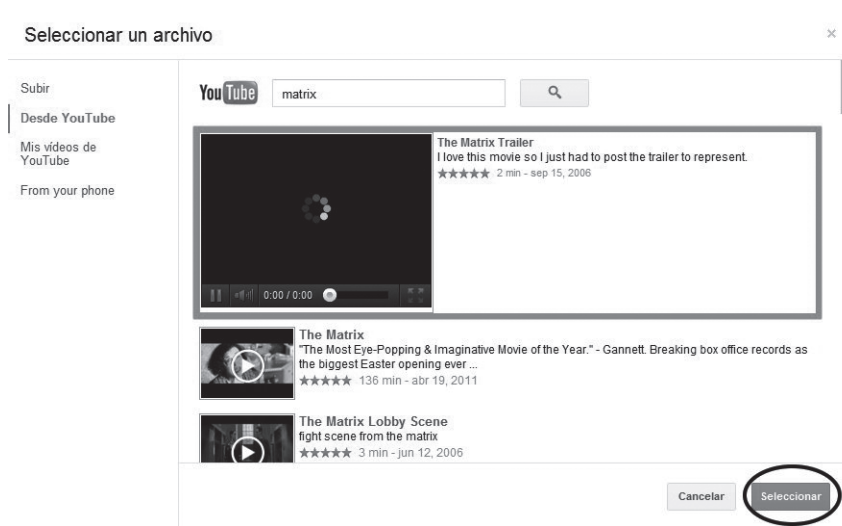


Figura 4.37. Seleccionar vídeo

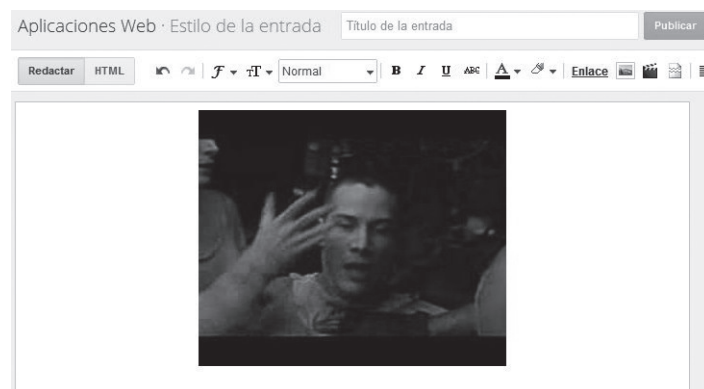


Figura 4.38. Publicar entrada

ACTIVIDADES 4.5



- Cree una nueva entrada en su *blog* incorporando un vídeo de YouTube con texto, como lo visto en la Práctica 4.2.

4.6.3 AÑADIR GADGETS A NUESTRO BLOG

Podemos añadir *gadgets* (añadidos) a nuestro *blog* de forma muy sencilla. En primer lugar, nos dirigimos a **Diseño** y se nos presentarán las zonas de nuestro *blog* en las que podemos añadir *gadgets* (podemos modificar estas zonas con el diseñador de plantillas).

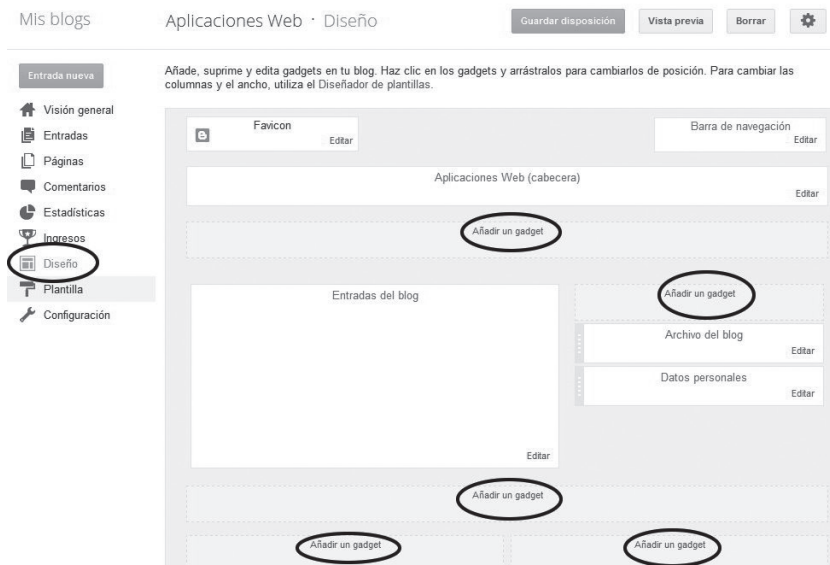


Figura 4.39. Diseño, añadir gadget

Tan solo debemos pulsar sobre **Agregar un gadget** y nos aparecerá una ventana con los **gadgets** disponibles para que seleccionemos el que más se ajuste a nuestras necesidades. Vamos a añadir un *gadget* consistente en una **Barra de vídeo**.



Figura 4.40. Seleccionamos un gadget

Título

YouTube ☐ Vídeos más vistos
☐ Vídeos mejor valorados
☐ Vídeos recientes

Canales
Nota: puedes especificar un único canal (por ejemplo, chicasolitaria115) o una lista de canales separados por comas

Palabras clave
Nota: puedes indicar una única palabra clave o una lista de palabras clave separadas por comas

Vista previa



con la tecnología de 

Figura 4.41. Barra de vídeos

Podemos configurar la temática de los vídeos que serán mostrados, mostrará un fotograma del vídeo y si pulsamos sobre él se reproducirá dentro de nuestro *blog*. En nuestro caso hemos utilizado **Palabras clave** para que los vídeos mostrados sean de temática “aplicaciones web”. Una vez pulsamos sobre **Guardar**, nos aparecerá en nuestro *blog* dicha barra de vídeos. Si pulsamos sobre uno de ellos se nos reproducirá el vídeo dentro del *blog* (Figura 4.42).

Aplicaciones Web



Figura 4.42. Vista previa gadget

ACTIVIDADES 4.6



➤ Añada tantos *gadgets* como vea interesantes en su *blog* y haga capturas de todo el proceso.

4.6.4 CONFIGURACIÓN AVANZADA DE NUESTRO BLOG

Desde la pantalla principal de administración de nuestro *blog* nos vamos al enlace **Configuración**. Dentro de **Configuración** tenemos una serie de pestañas que vamos a comentar a continuación:

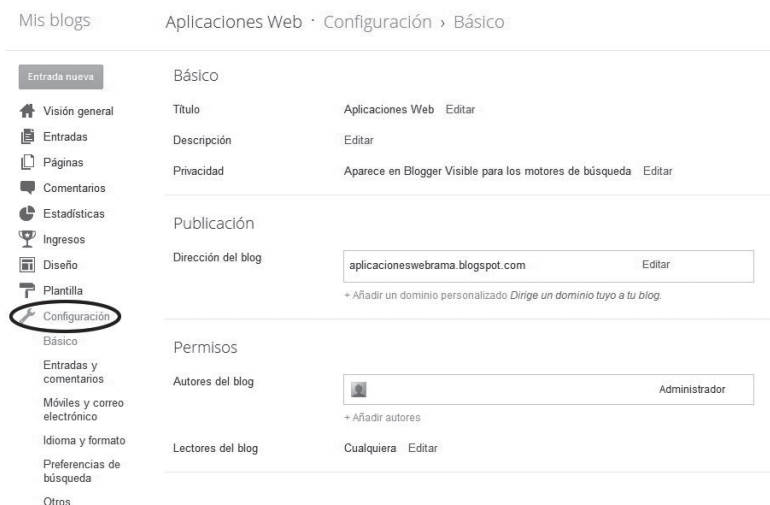


Figura 4.43. Configuración

- **Básico:** podemos cambiar el título al *blog*, añadir una descripción, permitir que el *blog* aparezca en las listas de Google, permitir ser encontrado por los motores de búsqueda, permitir la edición rápida de entradas en el *blog*, si el *blog* tiene contenido para adultos, etc.
- **Publicación:** podemos cambiar la dirección de nuestro *blog* (siempre y cuando la nueva esté disponible).
- **Permisos:** podemos añadir autores a nuestro *blog*, para que más usuarios puedan crear entradas en el mismo. También se puede limitar el acceso al *blog*, permitiendo solo a usuarios registrados y permitidos el acceso al mismo.
- **Entradas y comentarios:** en esta pestaña podemos cambiar el número de entradas a mostrar, el formato de la fecha y de la hora, etc. A su vez podemos permitir o no que se muestren los comentarios escritos, permitir qué usuarios pueden hacer comentarios en el *blog*, etc.
- **Móviles y correo electrónico:** direcciones que recibirán un *email* cuando se publique una nueva entrada en nuestro *blog*, dirección de correo para poder publicar entradas sin acceder directamente a Blogger y también dispositivos móviles que pueden crear entradas en el *blog* mediante SMS (mensajería móvil).

ACTIVIDADES 4.7



- Cambie la configuración del *blog* modificando varios de los aspectos que hemos visto en el apartado: aspectos básicos, de publicación, comentarios, etc. Hacer capturas de todo el proceso.

4.6.5 ESTADÍSTICAS DE NUESTRO BLOG

Al entrar a la configuración de nuestro *blog*, o cuando pulsemos sobre **Visión General**, se nos muestran las **estadísticas** del mismo. Con esto podemos conocer las estadísticas de visita de nuestro *blog*, el número de personas que han accedido, las ubicaciones aproximadas, número máximo de visitas en una fecha concreta, etc.



Figura 4.44. Estadísticas de nuestro blog

ACTIVIDADES 4.8

➤ Haga capturas de las estadísticas de su *blog*.

4.7 CÓMO PONER MÚSICA EN NUESTRO BLOG: GOEAR.COM

¿Le gustaría que los visitantes de su web y/o *blog* pudiesen escuchar la música que quiera? Con GoEar es muy sencillo.

Lo primero sería entrar en <http://www.goeear.com> y registrarse. El proceso gratis. Para ello pulsamos en **Registrarme**, que está en la parte superior de la ventana (ver Figura 4.45). Una vez registrados en el sistema, nos identificamos y accedemos a la página (Figura 4.46).

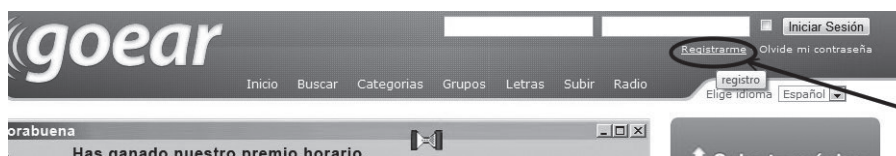


Figura 4.45. Registrarse en goear



Figura 4.46. Identificarse

Si la música que quiere publicar en su *blog* es muy conocida probablemente no tenga que subirla usted mismo y ya lo haya hecho alguien con anterioridad. Puede probar en la barra de búsqueda (Figura 4.47). En caso de que no exista en la base de datos debemos usar la opción **Subir** (Figura 4.48) y subiremos la canción nosotros mismos. Simplemente escribimos el título de la canción, el nombre del artista, álbum, el género o estilo de música y una pequeña descripción de lo que está subiendo. Seleccionamos el archivo y pulsamos **Subir** (Figuras 4.49 y 4.50).

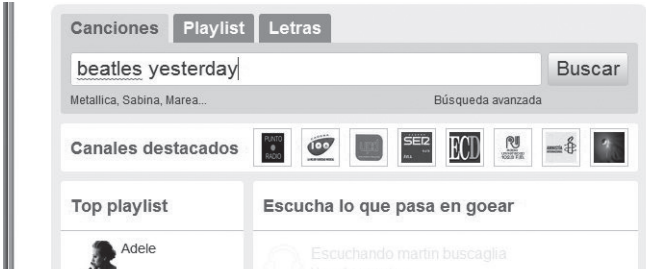


Figura 4.47. Búsqueda de música



Figura 4.48. Subir



Figura 4.49. Ficha de una canción



Figura 4.50. Subir una canción

Una vez hemos encontrado el archivo deseado o subido el archivo a *goear.com*, simplemente nos vamos a **Inserta esta canción en tu web**, copiamos el código que se nos muestra en pantalla y lo pegamos en nuestro *blog*.

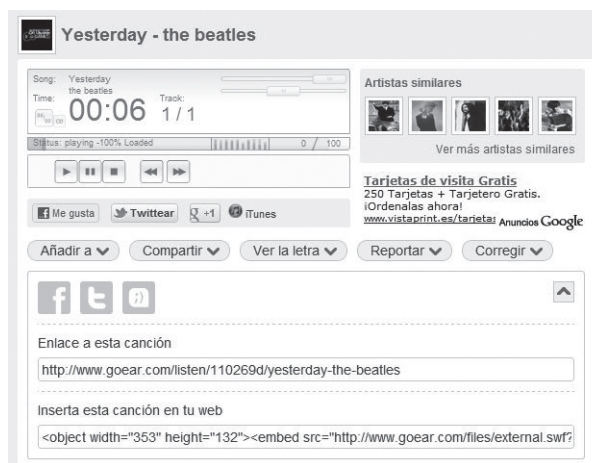


Figura 4.51. Inserta esta canción en tu web

Para insertar el código que hemos obtenido de *goear.com* en nuestro **blog** debemos pulsar sobre **Edición de HTML** (muy importante) y pegar dicho código.

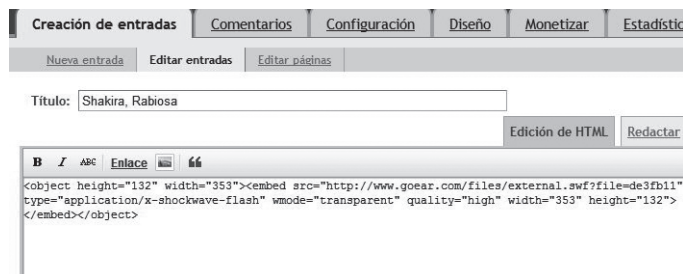


Figura 4.52. Insertar canción en nuestro blog

ACTIVIDADES 4.9



- Dese de alta en *goear.com* y publique una canción en su *blog* de la misma forma que se ha hecho en este apartado. Haga capturas de todo el proceso seguido.

4.8 CÓMO INSERTAR PRESENTACIONES A NUESTRO BLOG

La página web <http://issuu.com/> nos permite hacer algo parecido a goear.com, con presentaciones tipo slideshare (los usuarios pueden cargar archivos en los formatos de archivo: PowerPoint, PDF o OpenOffice presentaciones). El sitio web se considera que es similar a YouTube, pero para presentaciones de diapositivas.



Figura 4.53. issuu.com



Figura 4.54. Búsqueda de presentación



Figura 4.55. Seleccionamos una presentación

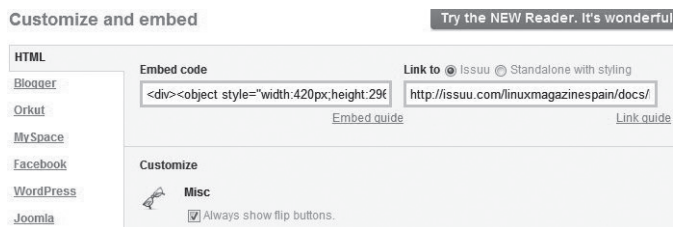


Figura 4.56. Obtenemos el código para nuestro blog

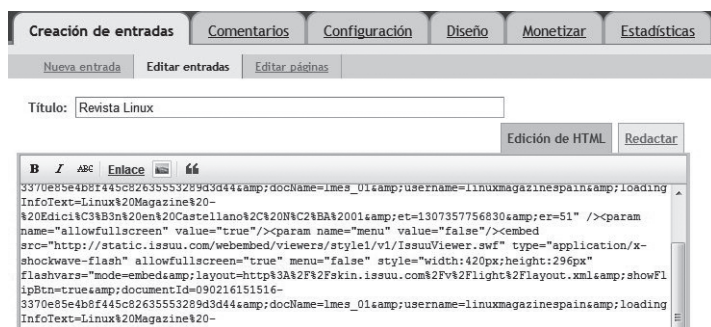


Figura 4.57. Insertar entrada como Edición HTML

Revista Linux



Figura 4.58. Listo

“ RECORDATORIO

No se olvide que debemos entrar en modo Edición HTML.

ACTIVIDADES 4.10

- Añada una presentación en su *blog* de la misma forma que se ha hecho en este apartado. Haga capturas de todo el proceso seguido.



RESUMEN DEL CAPÍTULO

En este capítulo hemos definido y establecido las partes de las que se compone un *blog*. Acto seguido hemos practicado con uno de los gestores de *blogs* más conocidos y utilizados en la actualidad: Blogger.

Hemos visto cómo crear una cuenta de correo electrónico y una cuenta Google, ya que ambas nos serán imprescindibles para poder elaborar nuestro *blog*.

Dentro de Blogger hemos visto cómo configurar la apariencia, cómo insertar entradas, fotos, vídeos e incluso cómo poner música en nuestro *blog* y presentaciones de tipo *slideshare*.



EJERCICIOS PROPUESTOS

- 1. Defina con sus palabras qué es un *blog*.
- 2. Enumere las diferentes partes de un *blog*.
- 3. ¿Conoce algún *blog* famoso?, ¿cuál?
- 4. ¿De qué herramientas dispone Google además de Blogger?
- 5. ¿Por qué cree que los *blogs* tienen tanta popularidad en la actualidad?
- 6. Haga una lista con 5 *blogs* que haya encontrado por la Red. Cada uno de los *blogs* debe ser de una temática distinta al resto, es decir, si el primero trata sobre la pesca, ya no se puede repetir el tema en los siguientes *blogs*.



TEST DE CONOCIMIENTOS

- 1 Un *blog* está formado principalmente por:
 - a) Cabecera.
 - b) *Posts*.
 - c) Comentarios.
 - d) Etiquetas.
 - e) *Widgets*.
 - f) Todas son ciertas.
 - g) Todas son falsas.
- 2 Blogger aparece en el año:
 - a) 1995.
 - b) 1997.
 - c) 1999.
 - d) 2003.
 - e) 2011.
- 3 Google se hace cargo de Blogger en el año:
 - a) 1995.
 - b) 1997.
 - c) 1999.
- d) 2003.
- e) 2011.
- 4 Las cuentas de correo de Gmail son:
 - a) Gratuitas.
 - b) Gratis las 3 primeras, a partir de la cuarta son de pago.
 - c) De pago.
 - d) De pago solo para empresas, para los usuarios comunes son gratis.
 - e) Todas son falsas.
- 5 En un *post* podemos insertar:
 - a) Textos.
 - b) Imágenes.
 - c) Sonidos.
 - d) Vídeos.
 - e) Presentaciones.
 - f) Todas son ciertas.
 - g) Todas son falsas.

5

Wordpress

OBJETIVOS DEL CAPÍTULO

- ✓ Conocer qué es WordPress y su historia.
- ✓ Crear una cuenta WordPress.
- ✓ Crear un *blog* con WordPress dotándolo de contenido.

5.1 WORDPRESS

WordPress es un sistema de gestión de contenido enfocado a la creación de *blogs*. Desarrollado en PHP y MySQL, bajo licencia GPL y código modificable, tiene como fundador a Matt Mullenweg. WordPress fue creado a partir del desaparecido b2/cafeblog y se ha convertido junto a Movable Type en el CMS más popular de la blogosfera². Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos.

Otro motivo a considerar sobre su éxito y extensión, es la enorme comunidad de desarrolladores y diseñadores, que se encargan de desarrollarlo en general o crear *plugins* y temas para la comunidad, siendo usado en septiembre de 2009 por 202 millones de usuarios.

Con WordPress tenemos dos opciones: crearnos un *blog* en su plataforma comercial (*wordpress.com*) o en nuestro propio servidor gracias a su software bajo licencia GPL (*wordpress.org*). En ambos casos el gasto es cero pero con excepciones. En el primero, determinadas funcionalidades son de pago (por ejemplo, la posibilidad de editar el CSS de nuestra plantilla) y, en el segundo, el coste estará en el servidor en el que decidamos montar nuestro *blog*.

Es de código abierto, está escrito en PHP y usa MySQL como gestor de base de datos. Uno de sus puntos más fuertes es la enorme comunidad de desarrolladores que tiene a sus espaldas. Al ser software libre, cualquiera puede escribir un *plugin* para agregar una funcionalidad determinada: desde la más rara (que nos aparezcan frases del musical Hello, Dolly! en nuestro panel de administración) a la más avanzada. Por esta razón tenemos un gran repositorio de *plugins* y plantillas para personalizar tanto como queramos nuestra bitácora.

Como curiosidad queda que cada nueva versión recibe el nombre de una leyenda del jazz: Miles Davis, Art Blakey, Charles Mingus, Billy Strayhorn...

- **Pros de WordPress.org:** es software libre, lo que nos aporta una transparencia de su funcionamiento. Además, cuenta con una gran comunidad de desarrolladores que mantienen el sistema vivo y en continua evolución. El contenido es completamente tuyo dado que eres tú el que almacena los archivos y la base de datos del *blog*. Se tiene un control total sobre el diseño. Permite importar contenidos de otras plataformas como Blogger, LiveJournal, Movable Type... Se puede ampliar fácilmente la funcionalidad de este gestor de contenidos mediante extensiones.
- **Contras de WordPress.org:** somos los responsables de la instalación y del mantenimiento del software y WordPress se caracteriza por las continuas actualizaciones para mejorar versiones anteriores y arreglar los fallos que éstas pudieran tener. Se necesita disponer de alojamiento y de dominio. Las copias de seguridad corren de nuestra cuenta.
- **Pros de WordPress.com:** la plataforma evoluciona al mismo ritmo que su hermana libre (*WordPress.org*), lo que hace que sea un sistema que se adapta a las novedades que aparecen en la cambiante Red. Nos ahorramos los gastos de un servidor y un dominio propio, así como el trabajo que supone actualizar la versión del software.
- **Contras de WordPress.com:** si deseamos instalar *plugins* o hacer modificaciones sobre el diseño de nuestra bitácora, deberemos tirar de cartera. Lo mismo que para ampliar los 3 GB de espacio para el almacenamiento de imágenes y ficheros subidos. Nuestros datos están en servidores de “la nube”.

5.1.1 HISTORIA DE WORDPRESS

El desarrollador principal, Matt Mullenweg, eligió el nombre WordPress por sugerencia de su amiga Christine Selleck. Las versiones lanzadas de WordPress tienen como nombre en clave músicos de *jazz* como, por ejemplo, la versión 1.0, Mingus. La primera versión final de WordPress se lanzó oficialmente el 3 de enero de 2004. La versión 1.5, que fue lanzada a mediados de febrero de 2005, tiene como nombre en clave “Strayhorn” (Billy Strayhorn) y ofreció una gama amplia de nuevas características, como el uso de páginas estáticas.

Facilita la administración de páginas fuera del orden cronológico “normal” del *weblog* y ha sido el primer paso para transformarse de un software básico de administración de *blogs* a un completo sistema de administración de contenidos.

Otra funcionalidad es la del sistema de creación de plantillas (*Themes*), las cuales permiten al usuario activar o desactivar una u otra según deseen para sus sitios. WordPress también ha sido equipado con una nueva plantilla (con nombre en clave Kubrick denominada como *Default*), además de la tradicional e inicial *Classic*, sencilla y válida con los estándares del W3C.

5.2 DAR DE ALTA NUESTRO BLOG

Nos dirigimos a la página web <http://es.wordpress.com/> y nos aparecerá la pantalla de presentación de Wordpress. Si clicamos sobre **¡Regístrate Ahora!** podemos dar de alta nuestro nuevo *blog*.



Figura 5.1. Página de Wordpress

Rellenamos los datos para dar de alta el *blog*. Vamos a utilizar el nombre “webaplica” para nuestro *blog*.

Figura 5.2. Cuenta Wordpress

El nombre debe estar disponible, por tanto, la dirección de nuestro nuevo *blog* será *webaplica.wordpress.com*. Clicamos sobre **Sign up** (ver Figura 5.3). Debemos comprobar el correo para finalizar el registro.

Figura 5.3. Alta de nuestro blog

Check Your E-mail to Complete Registration

An e-mail has been sent to implantaciondeaplicacionesweb@gmail.com to activate your account. Check your inbox and click the link in the message. It should arrive within 30 minutes. Si no activas tu cuenta dentro de 2 días, tendrás que volver a darte de alta.

¡Actualiza tu Perfil!

If you haven't got your activation e-mail why not update your profile while you wait?

Nombre:	
Apellido:	
Acerca de ti:	

Figura 5.4. Fin de registro

Nuestro *blog* se ha creado satisfactoriamente. Solo nos falta el paso final, abrir el correo que se nos ha enviado a nuestra dirección y clicar sobre el enlace que se nos facilitará en dicho correo (Figura 5.5).

Pulsando sobre dicho enlace nos llevará a la pantalla inicial de nuestro nuevo *blog* (Figura 5.6).



Figura 5.5. Alta

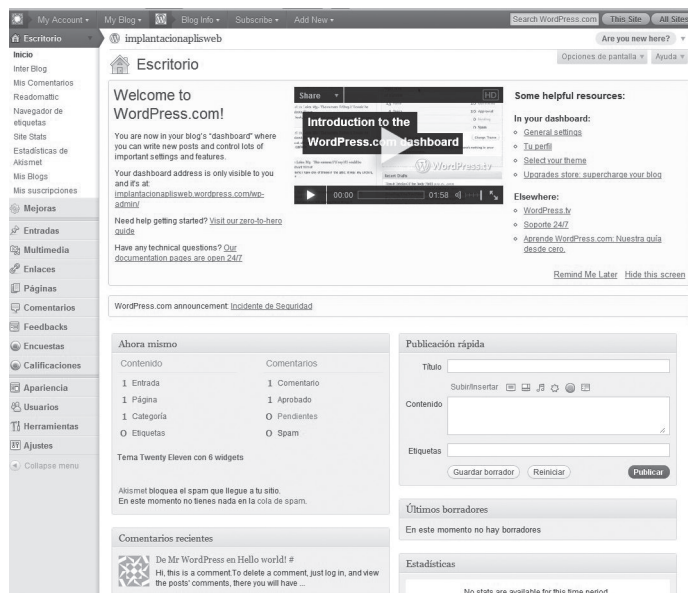


Figura 5.6. Alta

ACTIVIDADES 5.1

- ➔ Siguiendo las instrucciones del apartado, dese de alta en WordPress para crear nuestro nuevo *blog*. Haga capturas de todo el proceso.

5.3 TEMAS

Vamos a cambiar el tema que viene por defecto en nuestro *blog*. Para ello, dentro de la pestaña **Apariencia**, nos vamos a **Temas**, aquí nos van a aparecer multitud de temas distintos (habrá algunos que sean de pago). De entre los que nos aparecen podemos elegir el que más nos guste.

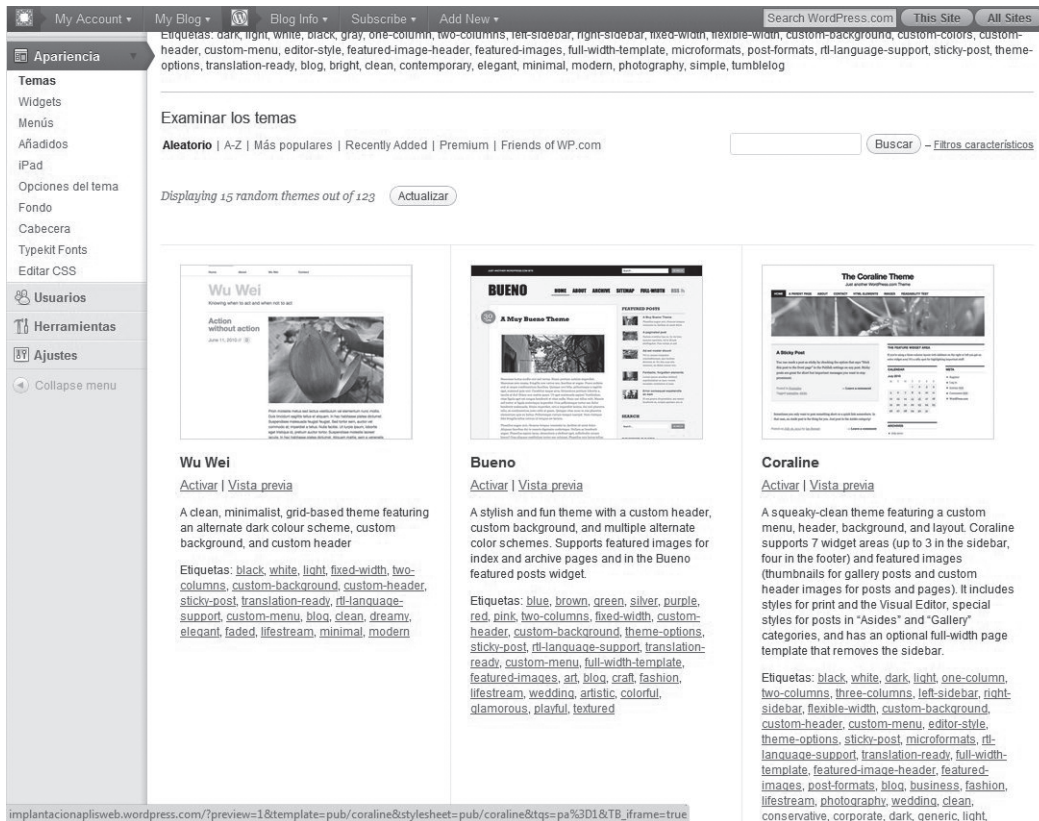


Figura 5.7. Temas

ACTIVIDADES 5.2



➡ Cambiar el tema de nuestro *blog* WordPress y hacer capturas de todo el proceso.

5.4 WIDGETS

Podemos agregar una serie de *gadgets* o *widgets* a nuestro *blog*. Estos están disponibles dentro del menú **Apariencia**, en la sección **Widgets** (Figura 5.8). Para incluir un *widget* en nuestro *blog* debemos arrastrar el *widget* seleccionado a la zona que identifica una parte de nuestro *blog* (*sidebar*, *left*, *middle*, *right* o *header*). En la Figura 5.9 podemos ver cómo hemos añadido el widget **Autores** a **Footer Left**.



Figura 5.8. Apariencia - Widgets

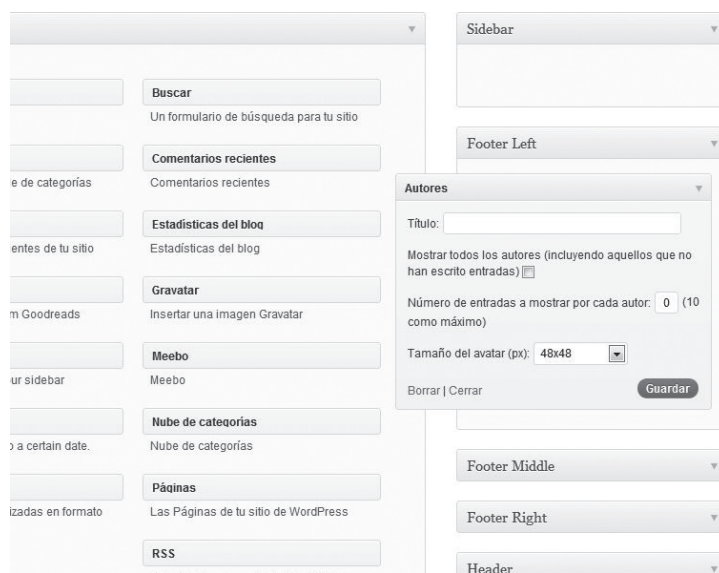


Figura 5.9. Añadimos un widget

ACTIVIDADES 5.3



➤ Añadir tantos *Widgets* como creamos oportunos a nuestro *blog* WordPress. Hacer capturas de todo el proceso.

5.5 FONDO

Podemos cambiar el fondo de nuestro *blog* por uno personalizado, para ello nos vamos al menú **Apariencia, Fondo** (Figura 5.10). Se nos da la posibilidad de seleccionar un archivo para que sea el nuevo fondo de nuestro *blog*.

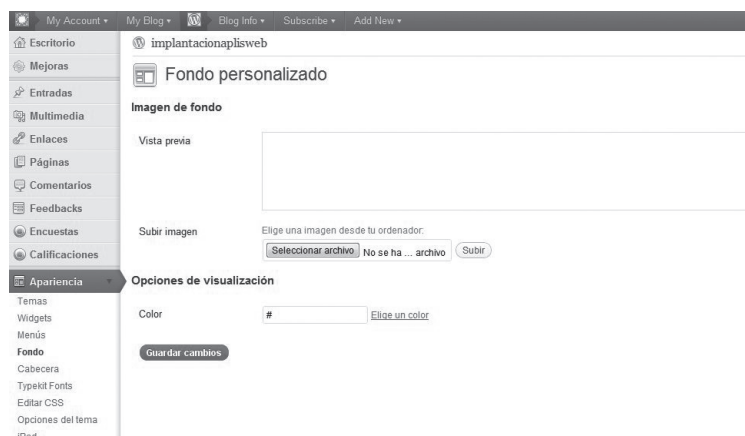


Figura 5.10. Apariencia - Fondo

Pulsamos en **Seleccionar archivo** y elegimos el archivo que será el nuevo fondo (Figura 5.11). Una vez elegido, pulsamos sobre **Subir** (Figura 5.12). Para que los cambios se lleven a cabo debemos clicar sobre **Guardar Cambios** (ver Figura 5.12). A partir de este momento nuestro *blog* llevará el nuevo fondo que hemos elegido (Figura 5.13).

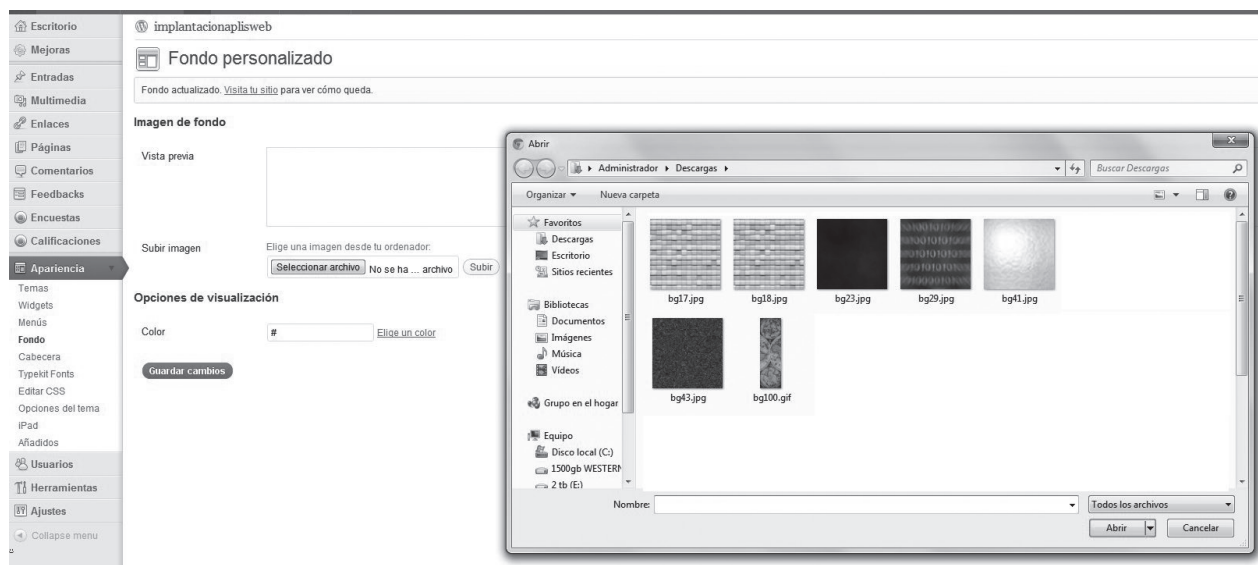


Figura 5.11. Seleccionar nuevo fondo

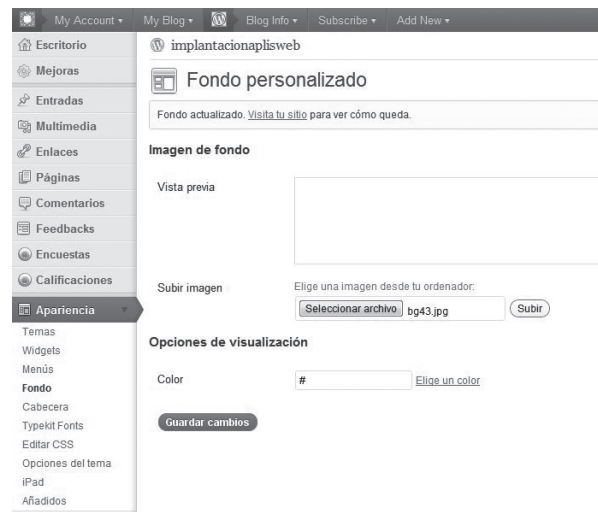


Figura 5.12. Guardar cambios

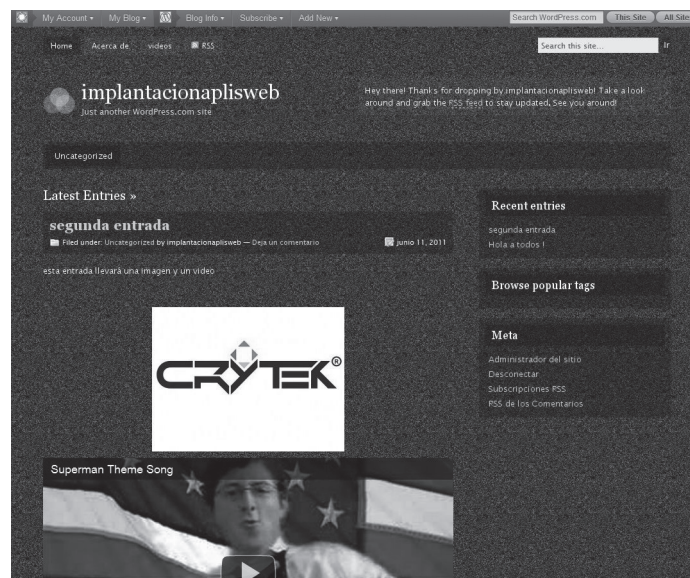


Figura 5.13. Nuevo fondo

ACTIVIDADES 5.4



- Cambiar el fondo a nuestro *blog* WordPress. Hacer capturas de todo el proceso.

5.6 CREACIÓN DE ENTRADAS EN NUESTRO BLOG

En WordPress, al igual en Blogger, podemos insertar entradas de texto, imágenes, vídeos, sonidos con *goear.com*, presentaciones con *issuu*, etc. A continuación, se presentan capturas de varios tipos de ellas empezando por entradas de texto, insertando imágenes y luego una mezcla de texto con imágenes.

Para crear entradas de texto nos vamos **Nueva Entrada** (Figura 5.14), podemos insertar un título al *post* (opcional) y el texto de la entrada (Figura 5.15).



Figura 5.14. Creación entrada de texto

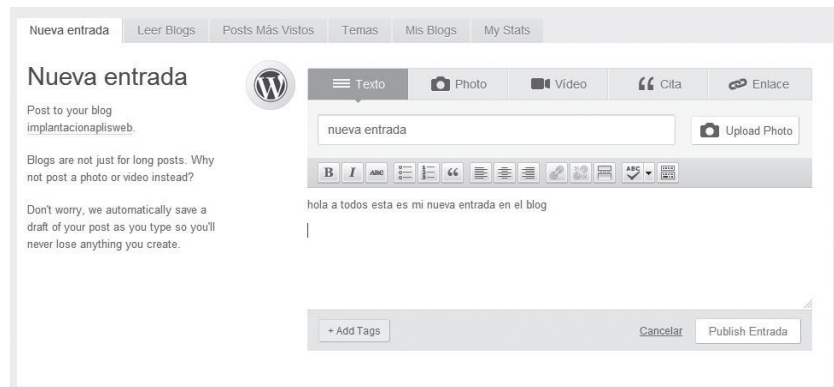


Figura 5.15. Creación entrada de texto

De forma similar podemos insertar una **Nueva entrada** con fotos (**Photo**), vídeos, citas y enlaces.

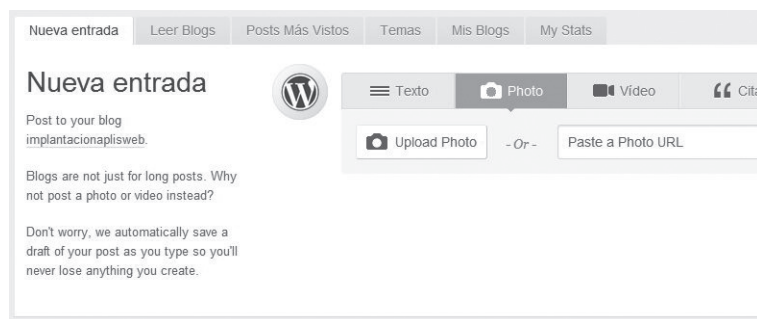


Figura 5.16. Creación entrada foto

Para crear un *post* con una foto, la foto en cuestión podemos subirla desde nuestro equipo o pegarla desde otro *blog* o web. En nuestro caso seleccionamos una foto que tenemos guardada en el equipo y la subimos (Figura 5.17).

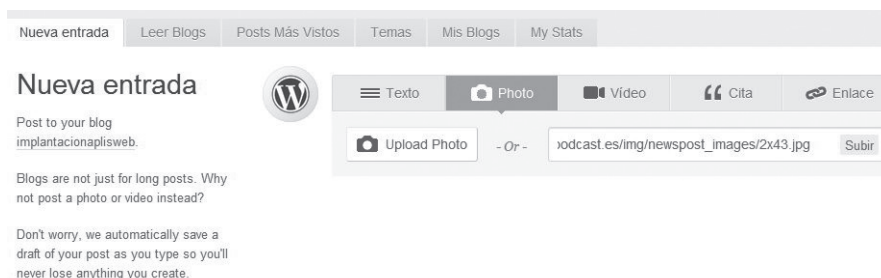


Figura 5.17. Creación entrada foto

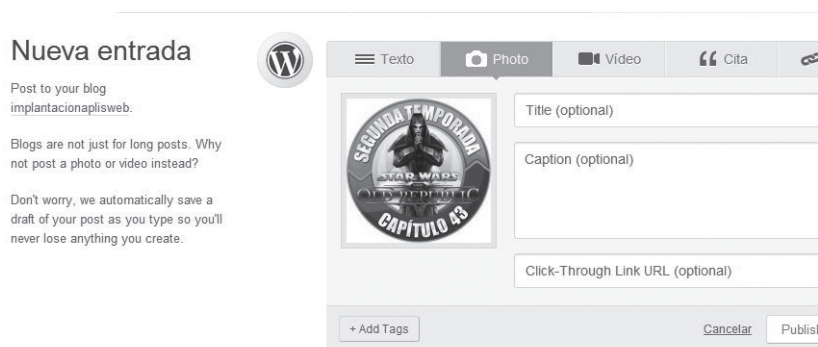


Figura 5.18. Subir la foto

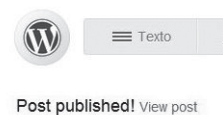


Figura 5.19. Publicar entrada

Podemos hacer también una mezcla de texto y fotos (como hicimos ya en Blogger). Para ello insertamos una nueva entrada de texto (Figura 5.20), introducimos el texto pertinente y utilizamos el botón **Upload Photo** que aparece al lado del título del *post* (Figura 5.21).

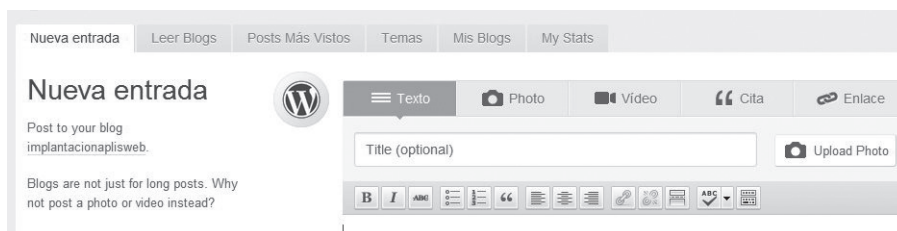


Figura 5.20. Nueva entrada de texto + foto(s)

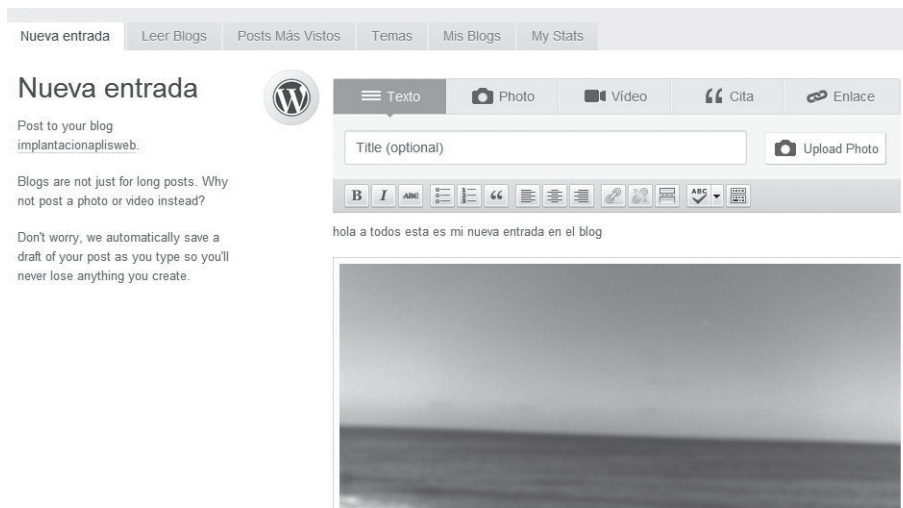


Figura 5.21. Entrada texto + foto

ACTIVIDADES 5.5



- Añadir una entrada de texto a nuestro *blog* WordPress y hacer capturas de todo el proceso.
- Añadir una entrada de imagen a nuestro *blog* WordPress y hacer capturas de todo el proceso.
- Añadir una entrada de texto con imágenes a nuestro *blog* WordPress y hacer capturas de todo el proceso.
- Añadir una entrada con un vídeo nuestro *blog* WordPress y hacer capturas de todo el proceso.
- Añadir una entrada con la canción que hemos insertado en *goear.com* en el capítulo anterior en nuestro *blog* WordPress y hacer capturas de todo el proceso.
- Añadir una entrada con la presentación que incorporamos en el capítulo anterior a nuestro *blog* WordPress y hacer capturas de todo el proceso.

5.7 ENCUESTAS

Una cosa interesante que tiene WordPress es la posibilidad de hacer encuestas entre los usuarios.

Para agregar una nueva encuesta a nuestro *blog*, nos dirigimos al enlace **Encuestas** (Figura 5.22) y pulsamos sobre **Crear una nueva encuesta** o **Create a Poll Now** (Figura 5.23).

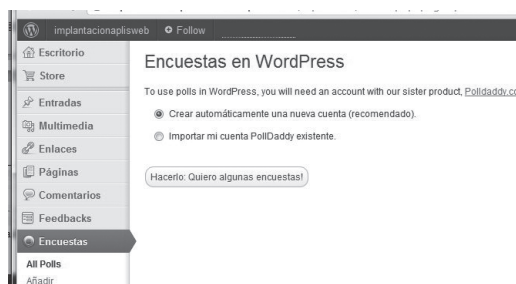


Figura 5.22. Encuestas

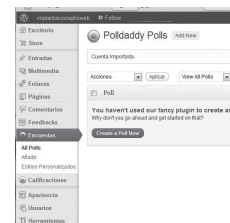


Figura 5.23. Create a new poll

Introducimos la pregunta de la encuesta y las opciones que damos a los usuarios. Podemos configurar las encuestas con selecciones multiples, estilos distintos (Figura 5.24), bloqueo de votos duplicados (para evitar irregularidades) y algunos detalles más (Figuras 5.25 y 5.26). En la Figura 5.27 se nos muestra cómo quedará nuestra encuesta dentro del *blog*.

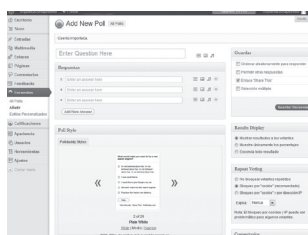


Figura 5.24. Nueva encuesta

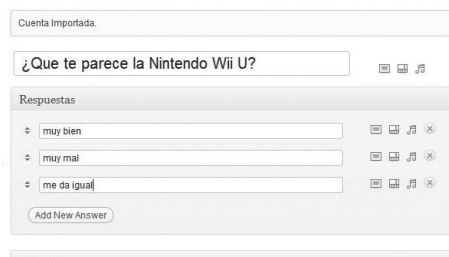


Figura 5.25. Cuestión y respuestas



Figura 5.26. Estilo en la encuesta



Figura 5.27. Vista en el blog

ACTIVIDADES 5.6

- Añadir una encuesta a nuestro *blog* WordPress y hacer capturas de todo el proceso.



RESUMEN DEL CAPÍTULO

En este capítulo hemos visto una de las herramientas para hacer *blogs* más utilizada (sobre todo por bloggers profesionales): WordPress. Hemos visto cómo darnos de alta y crear una cuenta en WordPress, ya que nos será imprescindible para poder elaborar nuestro *blog* en dicho gestor.

Dentro de WordPress hemos visto cómo configurar la apariencia, cómo insertar entradas, fotos, vídeos, además de crear encuestas sobre algún tema específico para conocer la opinión de los usuarios que visitan nuestro *blog*.



EJERCICIOS PROPUESTOS

- 1. ¿Qué es WordPress? ¿Para qué sirve?
- 2. ¿Qué diferencias encuentra entre Blogger y WordPress?
- 3. ¿Conoce algún *blog* WordPress famoso? ¿Cuál?
- 4. ¿Qué gestor de *blogs* le parece más sencillo, WordPress o Blogger? ¿Por qué?
- 5. ¿Qué diferencias encuentra entre *Wordpress.com* y *Wordpress.org*?



TEST DE CONOCIMIENTOS



1 WordPress nace en el año:

- a) 1999.
- b) 2002.
- c) 2004.
- d) 2006.
- e) 2011.
- f) Todas son ciertas.
- g) Todas son falsas.

2 Las principales características de WordPress son:

- a) Su licencia.
- b) Su facilidad de uso.
- c) Sus características como gestor de contenidos.
- d) Todas son ciertas.
- e) Todas son falsas.

3 WordPress incorpora una característica notoria que es:

- a) Subir vídeos.
- b) Subir fotos.
- c) Escribir *posts* de texto.
- d) Las encuestas.
- e) Todas son ciertas.
- f) Todas son falsas.

4 Las cuentas de WordPress son:

- a) Gratuitas.
- b) Gratis las 3 primeras, a partir de la cuarta son de pago.
- c) De pago.
- d) De pago solo para empresas, para los usuarios comunes son gratis.
- e) Todas son falsas.

5 En un *post* podemos insertar:

- a) Textos.
- b) Imágenes.
- c) Sonidos.
- d) Vídeos.
- e) Presentaciones.
- f) Todas son ciertas.
- g) Todas son falsas.

6

Wikis: MediaWiki

OBJETIVOS DEL CAPÍTULO

- ✓ Definir el concepto de wiki.
- ✓ Diferenciar los componentes de una wiki.
- ✓ Conocer y utilizar una de las wikis más utilizadas: MediaWiki.

6.1 DEFINICIÓN

Wiki proviene del hawaiano wiki wiki, “rápido”. Es un sitio web que puede ser editado por varios usuarios. Estos usuarios pueden crear, modificar y borrar el contenido de una página web, de forma fácil, rápida y sencilla. Hay un historial de recuperación, es decir, se puede volver a cualquier estado anterior de la página.

En 1995 se desarrolla la tecnología wiki para transmitir y compartir información en los grupos de informáticos implicados en el desarrollo de software.

En 2001 nace la Wikipedia, la mayor enciclopedia colaborativa creada por Jimbo Wales y Larry Sanger.

6.2 COMPONENTES DE UNA WIKI

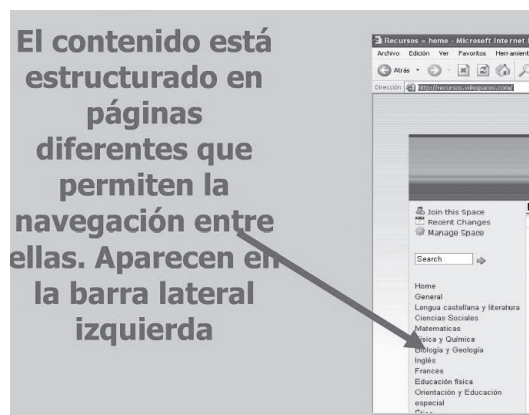


Figura 6.1. Navegación por la wiki

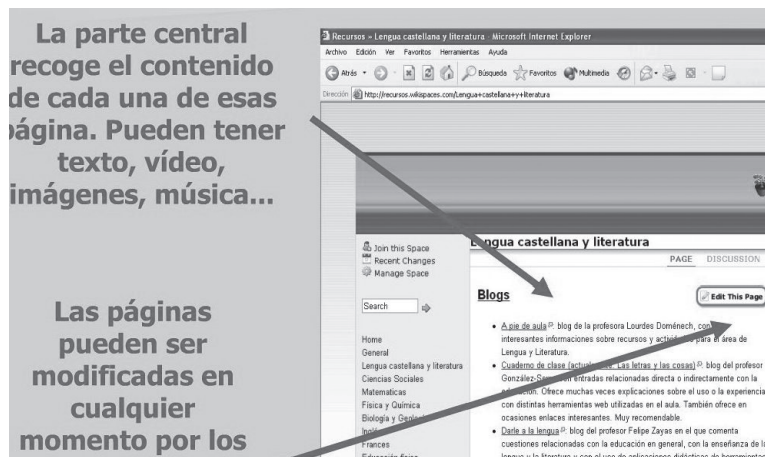


Figura 6.2. Contenido

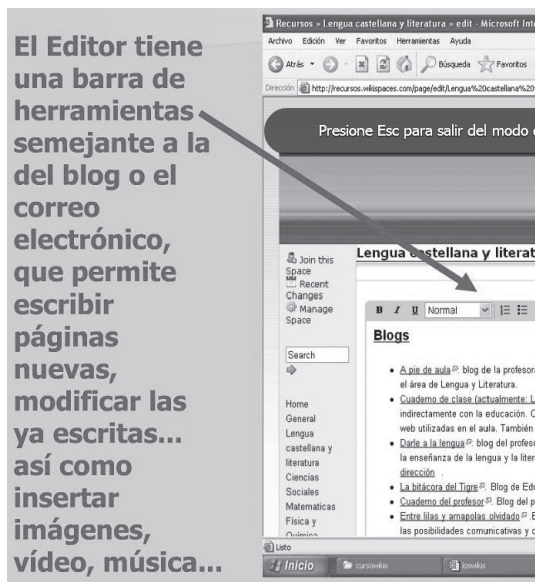


Figura 6.3. Editor

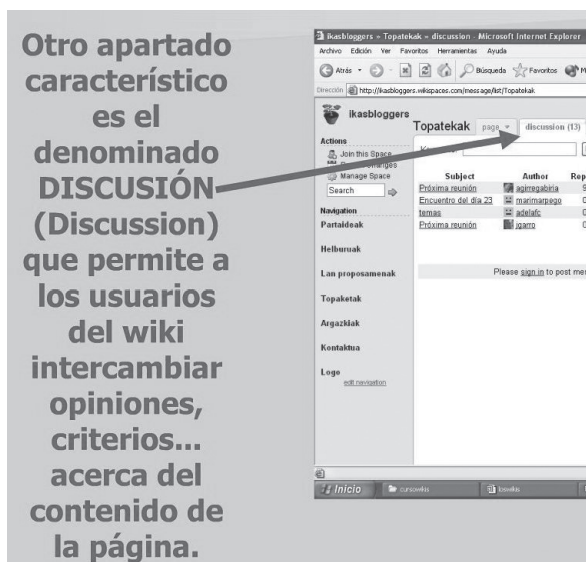


Figura 6.4. Discusión

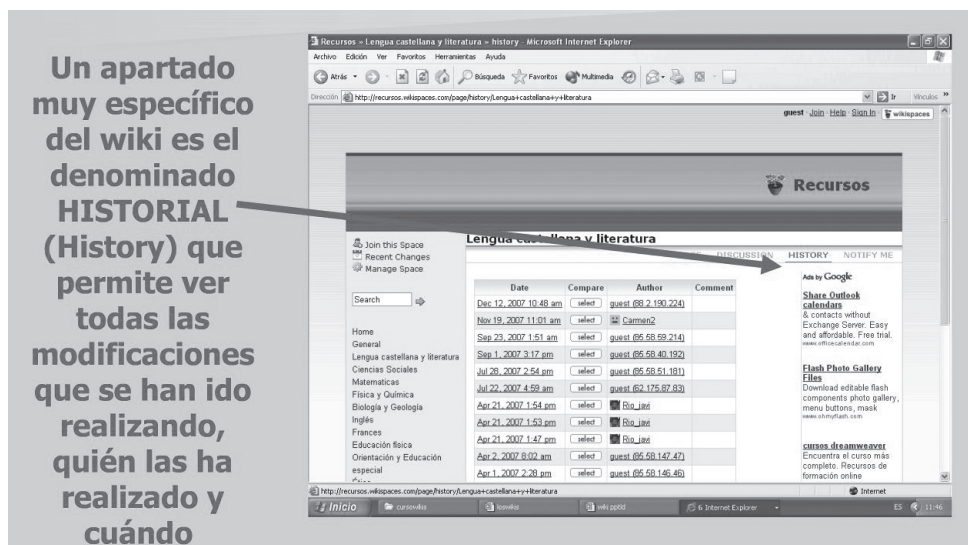


Figura 6.5. Historial

6.3 MEDIAWIKI

MediaWiki es un software para wikis libre programado en PHP. Es el software usado por Wikipedia y otros proyectos de la Fundación Wikimedia (Wikcionario, Wikilibros, etc.). Ha tenido una gran expansión desde el año 2005. Existen un gran número de wikis basados en este software que no mantienen relación con dicha fundación, aunque sí comparten la idea de la generación de contenidos de manera colaborativa.

MediaWiki puede ser instalado en los servidores web Apache e Internet Information Services y se puede usar como motor de base de datos MySQL o PostgreSQL.

Las características principales de MediaWiki son:

- ✓ A diferencia de las wikis clásicas, los nombres de las páginas no tienen por qué estar en CamelCase (mayúsculas mezcladas con minúsculas), lo que permite tener nombres más naturales.
- ✓ Espacios de nombres: permiten separar páginas de distintos tipos. Así, se puede tener un espacio de nombres para artículos, otro para plantillas, otro para debates, etc., que el software trata de distinta forma.
- ✓ Páginas de discusión: cada página de la wiki tiene una página de discusión propia.
- ✓ Soporte de TeX para visualizar fórmulas matemáticas.
- ✓ Listas de seguimiento, de tal forma que cada usuario pueda seguir los cambios en los artículos de su interés.
- ✓ Sistema de *plugins*, que permite extender fácilmente el software.
- ✓ Capacidad de bloquear temporalmente usuarios o páginas.
- ✓ Soporte de plantillas personalizadas con parámetros.
- ✓ Creación de líneas de tiempos a través de código wiki.
- ✓ Sistema de categorías jerárquico, que permite crear listados de artículos o de *thumbnails* de imágenes.
- ✓ Admite varios niveles de usuario, así como la posibilidad de que solo los usuarios registrados puedan editar, o de impedir el registro de más usuarios.
- ✓ Soporte para almacenamiento de memoria virtual o caché, también conocidos como *memcached* y el sistema de caché *Squid*.
- ✓ Plantillas (*skins*) personalizables por cada usuario.

6.4 LICENCIAS DE USO DE MEDIAWIKI

MediaWiki es un software libre de código abierto escrito originalmente para su uso en Wikipedia. Además de por la Wikipedia, es utilizado por otros proyectos de la Fundación Wikimedia y por muchas otras wikis, incluyendo el sitio web MediaWiki.

6.5 REQUERIMIENTOS DE FUNCIONAMIENTO

Para poder crear, mantener y explotar una wiki lo único que necesitamos es un navegador web y una conexión a Internet, independientemente de la plataforma utilizada.

6.6 INSTALACIÓN

El software MediaWiki está escrito en lenguaje PHP y utiliza como sistema de almacenamiento una base de datos. Los prerequisites para la instalación de la versión actual 1.X son los siguientes:

- Servidor web como Apache o IIS.
- Versión de PHP 5.0 o superior.
- Servidor de base de datos MySQL (versión 4.0 o superior) o PostgreSQL (versión 8.1 o superior).
- Es recomendable un administrador de la base de datos; PHPMyAdmin para MySQL o PHPPgAdmin para PostgreSQL.

Para descargar MediaWiki debemos dirigirnos a la web <http://www.mediawiki.org/wiki/Download>. Descomprimos la carpeta en el servidor web, en nuestro caso como es un servidor local, utilizaremos la carpeta **htdocs** de dicho servidor. Se recomienda cambiar el nombre de la carpeta descargada y descomprimida por MediaWiki para mayor comodidad a la hora de la configuración y uso, ya que la dirección será del tipo <http://servidor/mediawiki>.

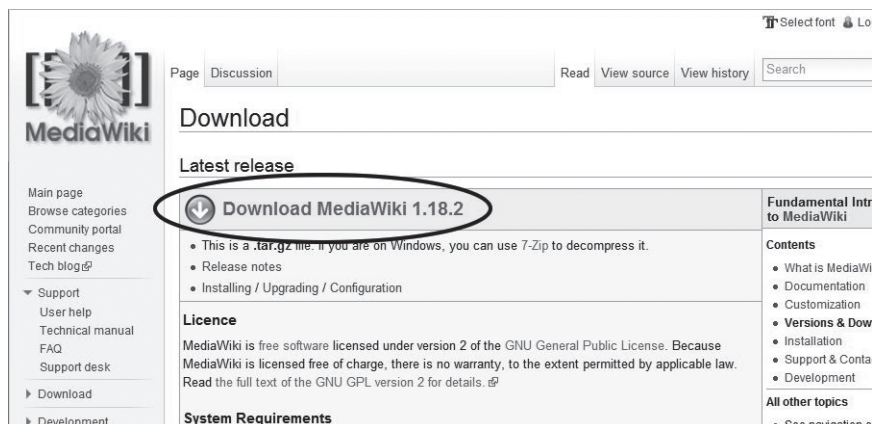


Figura 6.6. Descarga de MediaWiki

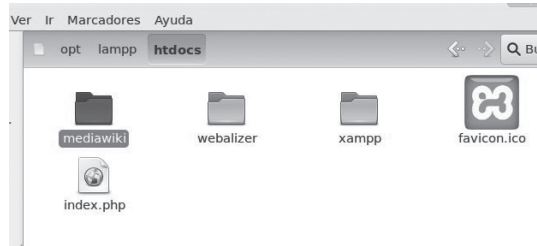


Figura 6.7. MediaWiki en htdocs

ACTIVIDADES 6.1



➤ Descargue MediaWiki de la web del creador. Haga capturas de todo el proceso realizado.

La instalación de MediaWiki tarda de 10 a 30 minutos. Se realiza mediante el *script* de instalación accesible a través del navegador web. Durante el proceso de instalación habrá que introducir diferente información relacionada con:

- Acceso a MediaWiki.
- Comprobación del entorno.
- Configuración del sitio.
- Configuración del *email*.
- Configuración de la base de datos.
- Completar la instalación.
- Problemas encontrados.

Los parámetros que se introduzcan durante la instalación pueden ser cambiados posteriormente en los ficheros de configuración.

6.6.1 ACCESO A MEDIAWIKI

Para acceder al *script* de instalación se escribe en el navegador web `http://miservidorweb/mediawiki`. Como se ha comentado anteriormente, al instalarlo en un servidor local la dirección de nuestra wiki será `http://localhost/mediawiki`



Figura 6.8. Instalación de MediaWiki

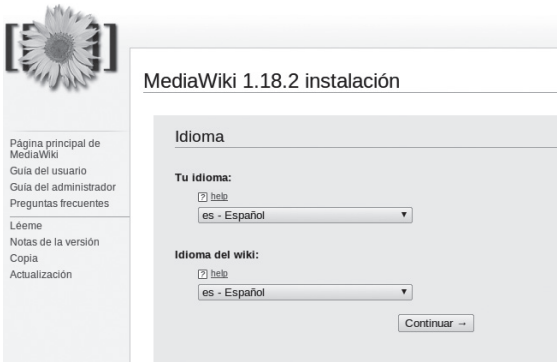


Figura 6.9. Selección idioma

6.6.2 COMPROBACIÓN DEL ENTORNO

Siguiendo el enlace que se muestra, MediaWiki procede a la comprobación del entorno, donde chequea que están disponibles todos los requisitos para que pueda funcionar correctamente como, por ejemplo, las versiones de PHP y de la base de datos.

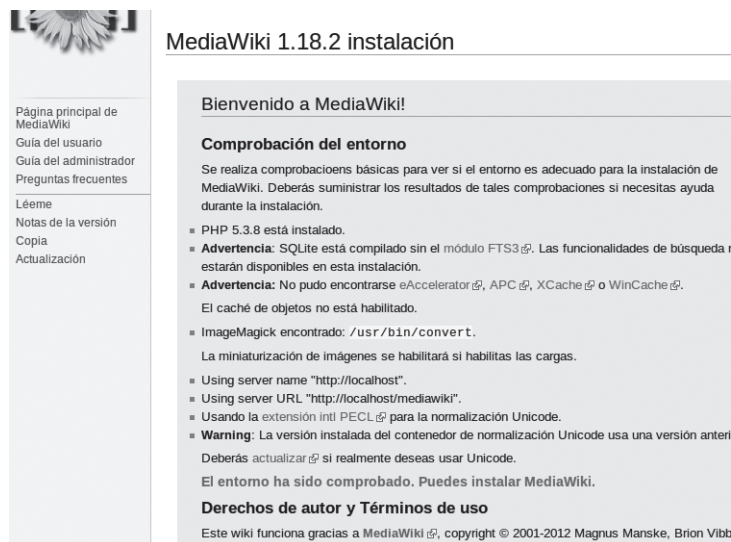


Figura 6.10. Comprobación del entorno

6.6.3 CONFIGURACIÓN DE LA BASE DE DATOS

A continuación se muestran los parámetros a configurar relacionados con la base de datos:

- **Tipo base de datos:** elegiremos el Sistema Gestor donde se instalará MediaWiki. Por defecto MySQL.
- **Servidor de la base de datos:** donde se encuentra la base de datos. En nuestro caso, como MediaWiki está instalada modo local, será *localhost*.
- **Nombre de la base de datos:** nombre de la base de datos, por defecto es *my_wiki*.
- **Prefijo para las tablas de la base de datos:** prefijo si se precisa acceder a una base de datos compartida con otras wikis o con otra aplicación web, para evitar conflictos.
- **Nombre de usuario de bases de datos:** nombre de usuario que se utilizará para la creación y el acceso a la base de datos.
- **Contraseña BD:** contraseña para el acceso del usuario anterior.
- **Cuenta de base de datos para acceso web:** podemos utilizar un usuario con menos privilegios para el desarrollo normal de nuestra Wiki. Por defecto toma los datos del usuario anterior.
- **Motor de almacenamiento:** puede seleccionarse InnoDB o MyISAM. En nuestro caso elegiremos *InnoDB* dado que soporta bien los accesos simultáneos.

- **Conjunto de caracteres de la base de datos:** podemos elegir entre almacenamiento binario y UTF-8. Con el método de almacenamiento binario MediaWiki almacena texto UTF-8 en campos binarios. Es más eficiente el método binario que el UTF-8, por tanto, seleccionaremos éste.

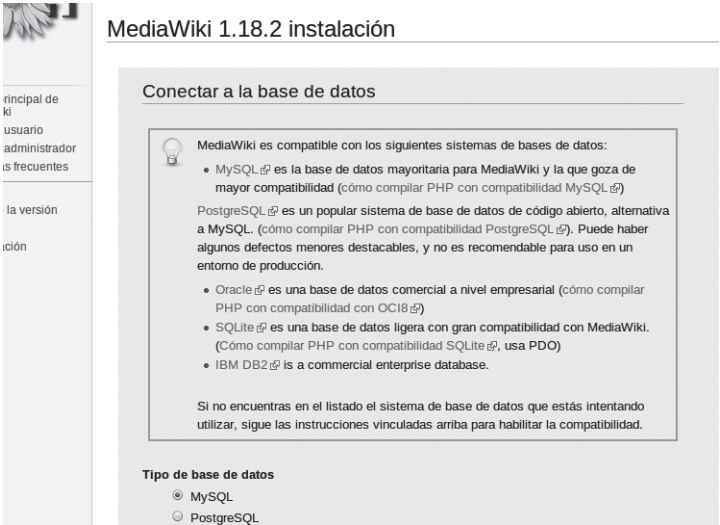


Figura 6.11. Base de datos

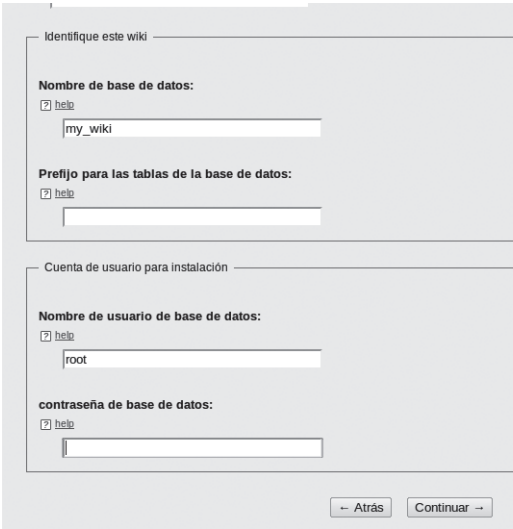


Figura 6.12. Base de datos



Figura 6.13. Base de datos

6.6.4 CONFIGURACIÓN DEL SITIO

Se introducirán parámetros relacionados con el sitio wiki:

- **Nombre del Wiki:** el nombre del sitio wiki.
- **Espacio de nombre de proyecto:** podemos indicar aquí un prefijo de espacio de nombre, así todas las páginas contendrán ese prefijo.
- **Cuenta de administrador:** nombre de usuario del administrador.
- **Contraseña:** contraseña del administrador de la wiki.
- **Dirección de correo electrónico:** una dirección de correo electrónico que nos permita recibir mensajes del resto de usuarios de la wiki, también nos puede permitir recibir correos frente a modificaciones de páginas supervisadas. No obstante podemos dejar este campo vacío.

Figura 6.14. Instalación

Figura 6.15. Instalación

6.6.5 OPCIONES DE INSTALACIÓN

- **Perfil de derechos de usuario:** aquí elegiremos el perfil de los usuarios de nuestra wiki. Podemos elegir entre **wiki tradicional** (cualquier persona puede colaborar, aún sin registrar), **Creación de cuenta requerida**, **solo editores autorizados** o **wiki privado**.
- **Copyright and licencia:** el contenido de lo que se publique en la wiki estará basado en esta licencia. Hay varias opciones: **Sin licencia** (el texto y las imágenes no tienen *copyright*, cualquiera podrá copiar y usar estos contenidos). **GNU free documentation license** (licencia de documentación libre de GNU, es la usada por Wikipedia.). **Licencia Creative Commons o CC** (de ésta hay de varios tipos), y otras. Puede encontrar más información sobre *Copyright* y licencias de MediaWiki en la siguiente página web: <http://www.mediawiki.org/wiki/Project:Copyrights>.

- **Activar el envío de e-mails** (global): por defecto, las notificaciones generales están habilitadas, como el envío de recordatorio de una contraseña o permitir el envío de *emails* entre usuarios.
- **Habilitar correo electrónico de usuario a usuario**: para permitir el envío de *emails* entre usuarios sin que se publiquen sus direcciones.
- **Activar notificaciones de páginas de discusión de usuarios**: para que se envíe un *email* cuando el usuario lo especifique sobre cambios en una página, o solo para las discusiones de página.
- **Activar autenticación del correo electrónico**: para que sea necesario introducir un *email* cuando se registre el usuario, además se debe confirmar la cuenta mediante *email*.

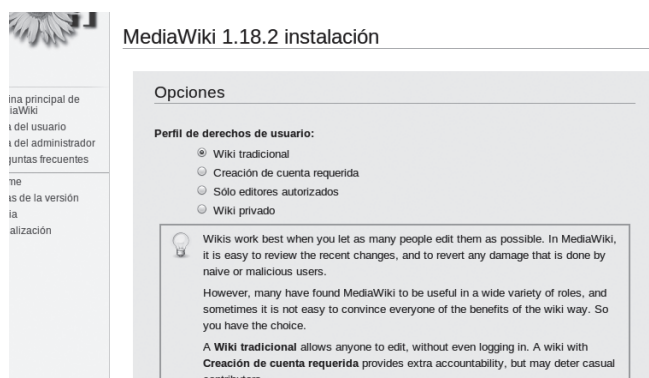


Figura 6.16. Perfil del usuario

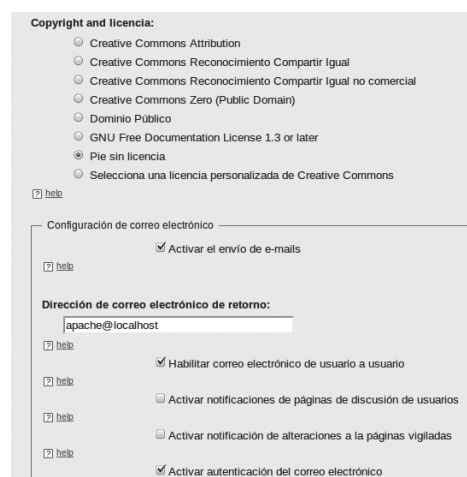


Figura 6.17. Copyright y correo

6.6.6 EXTENSIONES

La funcionalidad que ofrece por defecto MediaWiki se puede ampliar con un buen número de extensiones. Las que nos aparecen en la Figura 6.18 vienen incluidas directamente al descargar MediaWiki. Podemos encontrar muchas extensiones más en <http://www.mediawiki.org/wiki/Category:Extensions/es>.

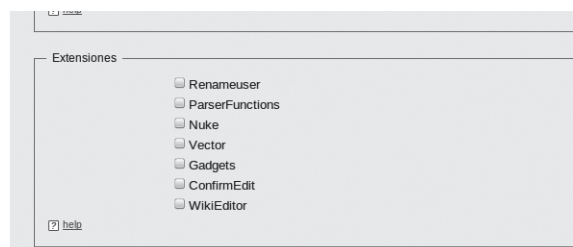


Figura 6.18. Extensiones

6.6.7 CARGA DE IMÁGENES Y ARCHIVOS. CONFIGURACIÓN AVANZADA

Si habilitamos esta opción podremos subir ficheros e imágenes a nuestra wiki, en caso de no habilitarla en esta ocasión podremos habilitarla después. Para ello debemos buscar en el fichero *Localsettings.php* la línea *\$wgEnableUploads* y cambiar *false* por *true*.

En caso de habilitar esta opción podremos elegir el **directorio para achivos eliminados** y la dirección **URL del logo**.

Con respecto a la configuración avanzada, utilizaremos **Sin almacenamiento en caché**, ya que no pretendemos realizar un sitio con millones de visitas concurrentes y **Utilizar Memcached** requeriría un esfuerzo extra que no es necesario en nuestro caso.

The screenshot shows the 'Cargas de imágenes y archivos' (Image and file uploads) section. It includes a 'Habilitar la subida de archivos' (Enable file uploads) checkbox which is checked. Below it, the 'Directorio para los archivos eliminados' (Directory for deleted files) is set to '/opt/lampp/htdocs/mediawiki/images/deleted'. The 'URL del logo' (Logo URL) is set to '/wiki/skins/common/images/wiki.png'. There is also a 'Habilitar Instant Commons' checkbox which is unchecked. The 'Configuración avanzada' (Advanced configuration) section shows 'Configuración de la caché de objetos' (Object cache configuration) with two radio buttons: 'Sin almacenamiento en caché' (No caching) which is selected, and 'Utilizar Memcached' (Use Memcached). At the bottom, there are 'Atrás' (Back) and 'Continuar' (Continue) buttons.

Figura 6.19. Imágenes y configuración avanzada

6.6.8 COMPLETAR LA INSTALACIÓN

Una vez se hayan introducido todos los parámetros requeridos se procederá a la instalación. Pulsamos sobre **Continuar** y, si algo ha fallado, muestra un error y vuelve a la misma pantalla para que se proceda a su corrección. Cuando está todo correcto, MediaWiki creará la base de datos, las tablas necesarias y el fichero de configuración.



Figura 6.20. Instalación

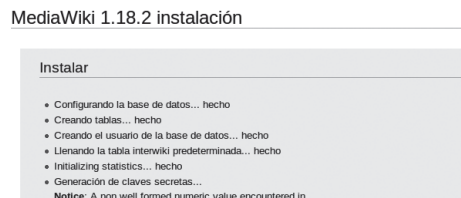


Figura 6.21. Finalizar instalación

Para completar la instalación, hay que copiar el archivo generado *localsettings.php* (Figura 6.23) al directorio raíz de la wiki (Figura 6.24). Ya podemos acceder a nuestra nueva wiki (Figura 6.25).

MediaWiki 1.18.2 instalación

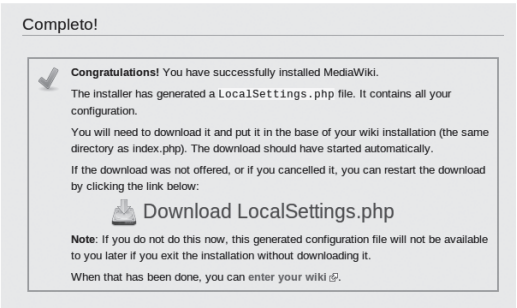


Figura 6.22. Descargar LocalSettings.php

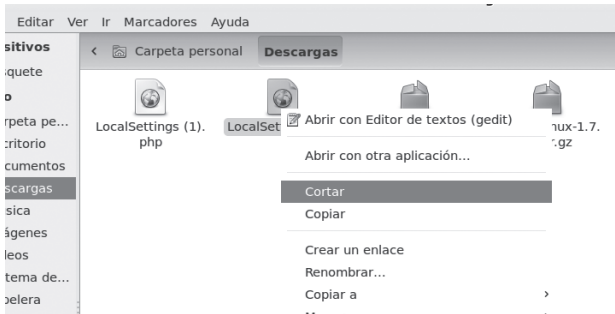


Figura 6.23. Copia fichero desde Descargas

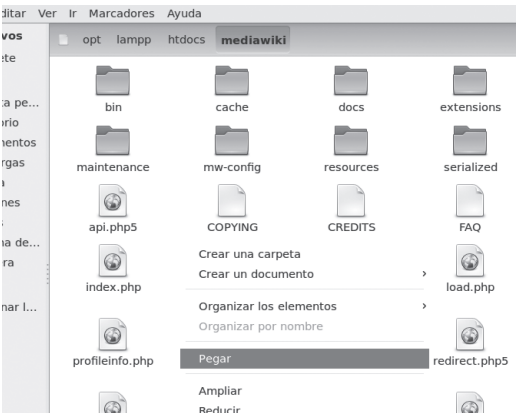


Figura 6.24. Copiamos a MediaWiki

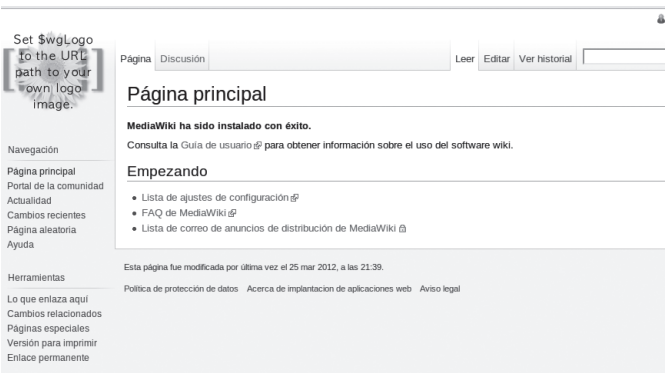


Figura 6.25. Nuestra nueva wiki

6.6.9 PROBLEMAS ENCONTRADOS

Cuando se introduce un parámetro que no es correcto, aparece un mensaje en rojo sobre el parámetro que causa problemas. Se debe corregir para continuar la instalación.

ACTIVIDADES 6.2

- Instale en su servidor local MediaWiki tal y como se ha realizado en el capítulo, cree la base de datos para utilizarla con MediaWiki. Haga capturas de todo el proceso realizado.

6.7 USUARIOS Y GRUPOS

Las wikis son creadas para ser libres y editables abiertamente. Esto significa que cualquiera puede editar las páginas, a menudo sin ni siquiera registrarse/identificarse. Esto solo es posible porque las wikis ofrecen un tipo de seguridad llamada “seguridad blanda”, por lo que es muy fácil de controlar las ediciones y revertir los cambios para recuperar una versión anterior de una página. En teoría no hay permisos. Todo el mundo es administrador. No se necesita ningún permiso para editar, de hecho, se anima a los usuarios a seguir adelante y participar. Por lo menos esa es la teoría. En la práctica, el software MediaWiki ofrece algunas opciones relacionadas con los permisos.

Para registrarnos en la wiki debemos pulsar sobre **Iniciar sesión/crear cuenta**, que está en la parte superior derecha de nuestro navegador (Figura 6.26). Después de esto nos mostrará una pantalla para iniciar sesión dando también la posibilidad de crear una cuenta nueva. Si pulsamos sobre **Crear una cuenta** nos pedirá una serie de datos: usuario, contraseña, correo y nombre real. Si rellenamos los datos correctamente estaremos registrados en la wiki y ya podremos acceder a las partes destinadas a *Usuarios registrados*.

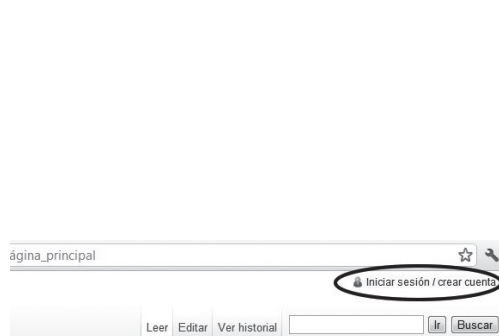


Figura 6.26. Iniciar sesión/crear cuenta

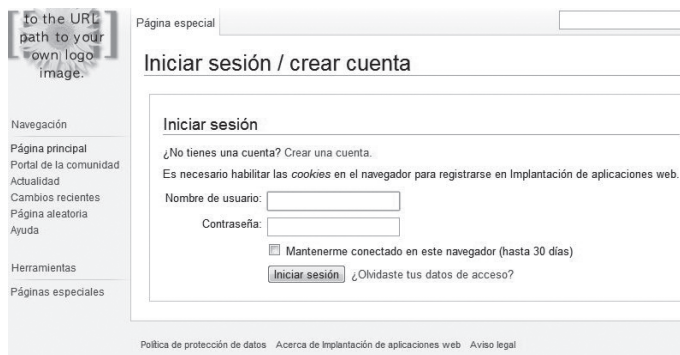


Figura 6.27. Iniciar sesión

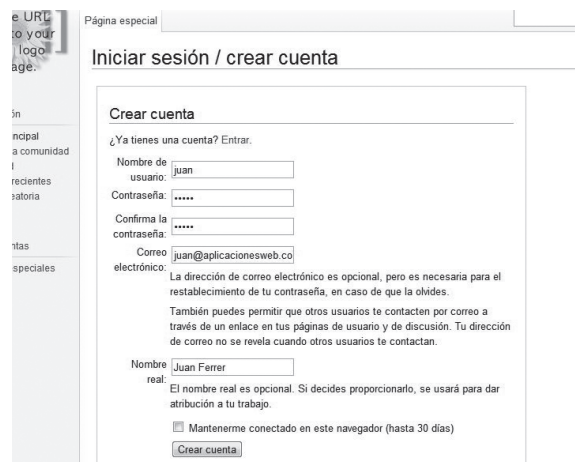


Figura 6.28. Crear cuenta

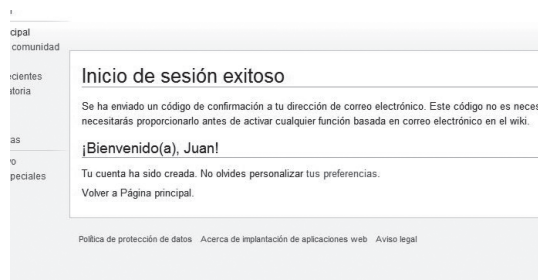


Figura 6.29. Inicio de sesión

6.7.1 SYSOPS O ADMINISTRADORES

MediaWiki ofrece una funcionalidad adicional para un grupo de usuarios de confianza llamados **sysops** (o más conocido generalmente como “administradores”). No debemos abusar de este tipo de usuarios en nuestra wiki, ya que un usuario normal puede participar en la edición de páginas e incluso realizar una reorganización drástica de la wiki. Los privilegios extra de un usuario *sysop* estarán destinados para su uso en la solución de conflictos y en la prevención de actos vandálicos.

Hay una serie de características adicionales disponibles para los operadores del sistema. Los más importantes son las siguientes:

- **Proteger páginas, desproteger páginas y editar páginas protegidas.** La “protección” de una página evita que los usuarios normales puedan editar la página.
- **Eliminación de páginas y restauración versiones antiguas.** Un usuario normal no tiene este permiso.
- **Bloqueo de un usuario o la dirección IP y desbloqueo de bloqueados.** Los usuarios pueden ser bloqueados desde la wiki. Esta es una forma útil de mantener alejados a los vándalos.
- **Supervisar y ver ediciones supervisadas.** Deben coordinarse con el resto de administradores, para asegurarse de que cada edición de la wiki ha sido revisada por un usuario de confianza.

6.7.2 LOS BURÓCRATAS

Otro nivel de permisos dentro de un MediaWiki es el “burócrata” del grupo. Lo que puede hacer un burócrata que un *sysop* no puede es cambiar son los permisos de usuarios básicos para hacerlos administradores del sistema y degradar el nivel de los operadores del sistema a usuarios normales.

Para cambiar los permisos a los usuarios, en la barra lateral aparece el cuadro **Herramientas**. Dentro debemos pulsar sobre **Páginas especiales** (Figura 6.30) y, dentro de esta opción, sobre **Usuarios y permisos** (Figura 6.31).

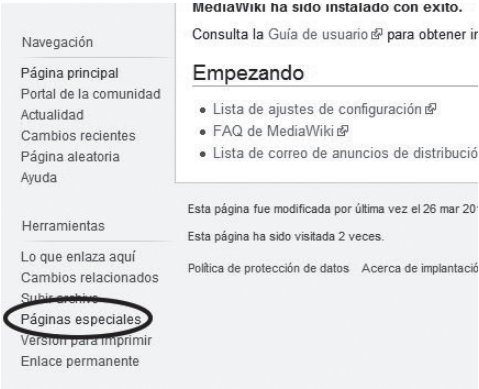


Figura 6.30. Herramientas, páginas especiales

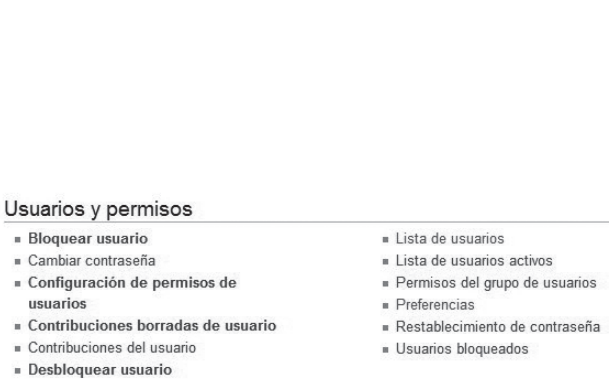


Figura 6.31. Usuarios y permisos

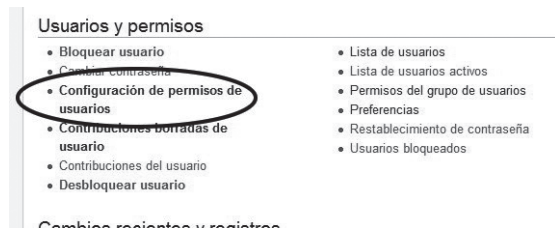


Figura 6.32. Configuración de permisos de usuarios

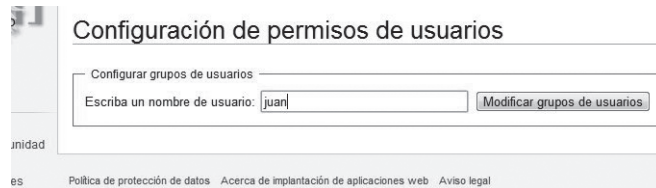


Figura 6.33. Buscar usuarios

Ahora nos vamos sobre **Configuración de permisos de usuarios**, a continuación buscamos el usuario al que vamos a cambiar los permisos (Figura 6.33).

Nos aparecerá el usuario que tenga ese nombre de usuario (si es que hay alguno), en nuestro caso asignamos el permiso de **administrador** al usuario de la búsqueda (Figura 6.34) y pulsamos sobre **Guardar grupos de usuarios**. A partir de este momento el usuario **juan** será del grupo de los **administradores** (Figura 6.35).

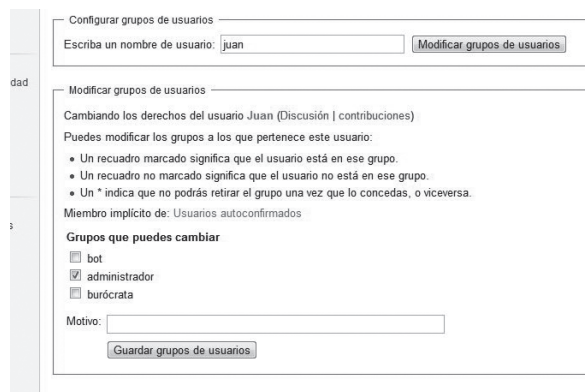


Figura 6.34. Asignar privilegios de administrador

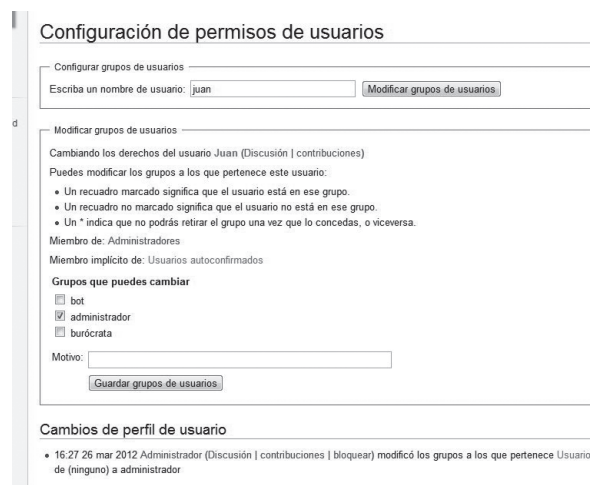


Figura 6.35. Privilegios cambiados

ACTIVIDADES 6.3



- Cree varios usuarios en su MediaWiki y después asígneles diversos roles (administrador, burócrata). Haga capturas de todo el proceso realizado.

6.8 CREACIÓN DE CONTENIDOS

Al ver cualquier página de MediaWiki vamos a encontrar tres principales elementos de navegación:

- ✓ La barra lateral le da acceso a páginas importantes en la wiki, como los cambios recientes o subir archivos. MediaWiki requiere que esté identificado en el sistema para poder ver todas las opciones de la barra lateral.
- ✓ Enlaces (a menudo llamados *tabs*) que se refieran a la página visualizada: su página de discusión, el historial de versiones y, sobre todo, el enlace.
- ✓ Enlaces de usuario, como un usuario anónimo. Verá un enlace para crear una cuenta o entrar como un usuario registrado. Usted tiene una colección de vínculos personales, incluyendo los de su página de usuario y preferencias.

6.8.1 SIDEBAR

La barra lateral se muestra en el borde izquierdo de la página, debajo del logo del sitio (si se utiliza la piel MonoBook por defecto). Esta barra lateral le da acceso a páginas importantes en la wiki, como los cambios recientes o subir archivos.

- **Navegación.** Al hacer clic en el logo volvemos a la página principal de la wiki. Los enlaces en la sección de navegación solo llevarán a páginas importantes de la wiki. Estos enlaces pueden ser configurados por los administradores del sitio.
- **Herramientas.** La caja de herramientas contiene una selección de enlaces que cambian en función del tipo de página que está viendo.

Hay unas herramientas que van a aparecer en las páginas, excepto en páginas especiales:

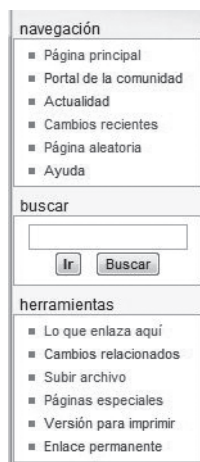


Figura 6.36. Sidebar

- **Lo que enlaza aquí:** le lleva a una página especial que enumera las páginas de esta wiki y contienen un enlace a la página actual. Esto es útil cuando usted está buscando para las páginas de información relacionada. La información también puede ser útil cuando se reestructuran páginas wiki y la necesidad de comprobar si los enlaces a esta página se mantienen después de los cambios en la página actual.
- **Cambios relacionados:** muestra los últimos cambios en las páginas que enlazan con la página actual.

Hay una serie de herramientas que van a aparecer en todas las páginas, incluyendo páginas especiales:

- **Subir archivo:** muestra una página especial que permite a los usuarios registrados subir imágenes y otros archivos en la wiki. Los archivos subidos pueden ser enlazados o incrustados en las páginas wiki. Esta opción está deshabilitada por defecto en MediaWiki. Para habilitarla debemos buscar en el fichero *Localsettings.php* la línea `$wgEnableUploads` y cambiar **false** por **true**.
- **Páginas especiales:** contiene información sobre la wiki y/o permite el acceso a las actividades de administración de la wiki.
- **Version para imprimir:** nos mostrará la página optimizada para su impresión (normalmente sin imágenes).
- **Enlace permanente:** se usa en los *blogs* para asignar una URL permanente a cada entrada de la wiki, para luego poder referenciarla, es decir, se facilita el acceso (y una referencia) directo a todos aquellos que puedan estar interesados en el contenido referenciado.

ACTIVIDADES 6.4



- Haga capturas del *sidebar* de su wiki. Habilite **Subir archivo**. Haga capturas de todo el proceso.

6.8.2 PESTAÑAS



Figura 6.37. Pestañas

Las pestañas se muestran en la parte superior de la página, a la derecha del logo del sitio (depende del *skin* utilizado). Estas pestañas permiten realizar acciones o ver las páginas que están relacionadas con la actual. Las acciones por defecto disponibles incluyen: visualización, edición y discusión de la página actual. Las pestañas específicas mostradas en sus páginas dependerán de si está o no registrado a la wiki y si usted tiene privilegios de administrador en la wiki. Los administradores pueden añadir o eliminar pestañas usando JavaScript o la instalación de extensiones, por lo que las pestañas pueden ser diferentes en función de la wiki que esté usando.

ACTIVIDADES 6.5



- Haga capturas de las pestañas de página de su wiki, vaya cambiando de usuario con distintos privilegios para ver las distintas pestañas.

6.8.3 ENLACES DEL USUARIO

Los enlaces se muestran en la parte superior derecha de la página (puede variar según el *skin* utilizado). Estas pestañas permiten al usuario registrado ver y editar su página de usuario y las preferencias wiki. Además, los enlaces de usuario también permiten que el usuario pueda acceder rápidamente a sus aportaciones de la wiki y cerrar la sesión. Para los usuarios anónimos los enlaces de usuario se sustituyen por un enlace a la página de inicio wiki o, si está activado, un enlace a su dirección IP y la página de su dirección IP.

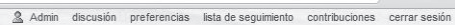


Figura 6.38. Enlaces del usuario

- **Nombre del usuario.** Enlaza con su página de usuario, donde puede mostrar información suya, información que desea recordar o cualquier cosa que le apetezca.

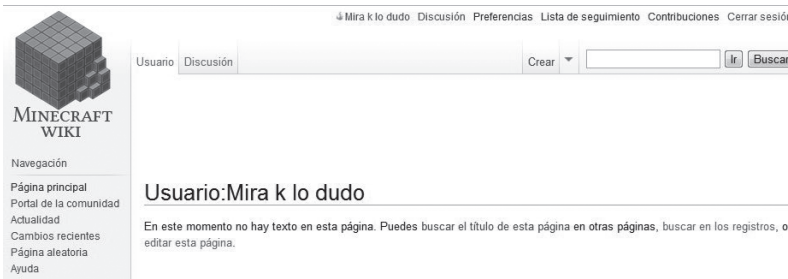


Figura 6.39. Nombre del usuario

- **Discusion.** Enlaza con su página de discusión, donde la gente puede dejarle mensajes.
- **Preferencias.** Aquí podemos configurar las preferencias para nuestro usuario. Podemos configurar una gran variedad de valores, como el *Perfil de usuario*, la *Apariencia*, *Fecha y hora*, *Edición* y muchos parámetros más.

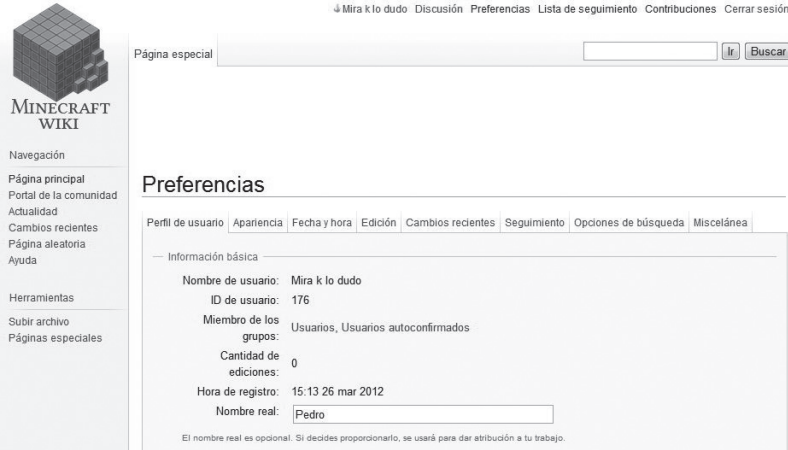


Figura 6.40. Preferencias

- **Lista de seguimiento.** Una lista de todas las páginas que está viendo.

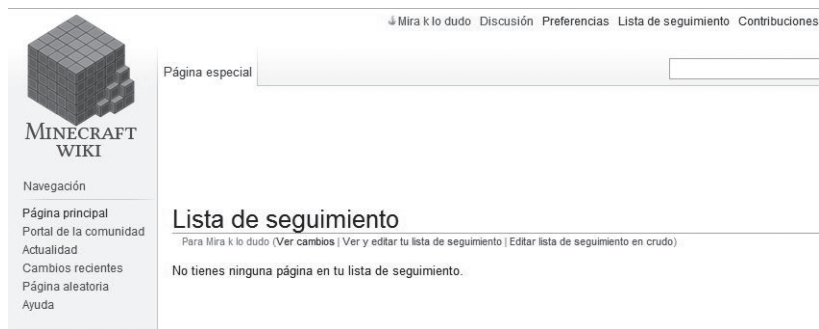


Figura 6.41. Lista de seguimiento

- **Contribuciones.** Una lista de todas las contribuciones que han hecho en la wiki.

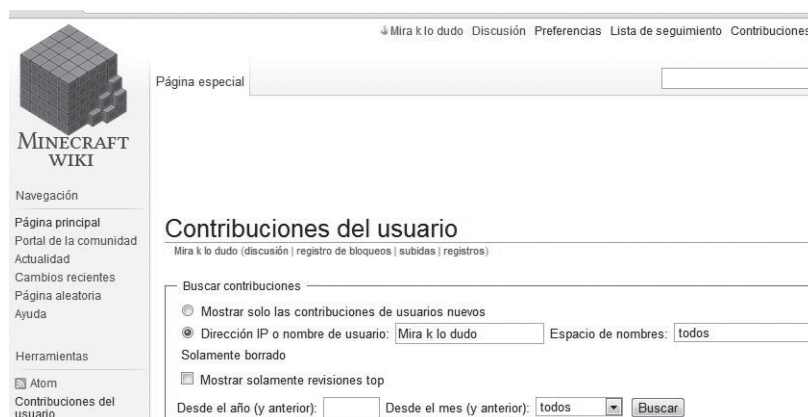


Figura 6.42. Contribuciones del usuario

- **Cerrar sesión.** Para desconectarse de la wiki.

ACTIVIDADES 6.6



- Haga capturas de los enlaces de usuario de su wiki. Cierre sesión y haga capturas de cómo quedarían los enlaces de un usuario sin identificar.

6.8.4 CREAR UNA PÁGINA NUEVA

Hay varias formas de crear una página nueva. Estas pueden variar en función del tipo de página a crear, así como de la wiki y del espacio.

MediaWiki hace que sea muy fácil para enlazar páginas utilizando una sintaxis estándar. Podemos crear un enlace a un artículo que aún no existe clicando sobre un enlace interno (Figura 6.43) y escribiendo el nombre de la nueva página (Figura 6.44), guardamos la página (Figura 6.45). El enlace será de color rojo (Figura 6.46).

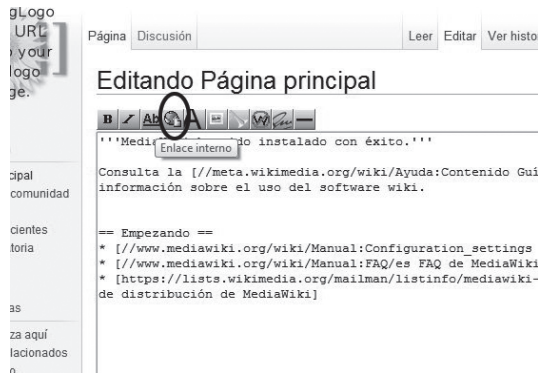


Figura 6.43. Enlace interno

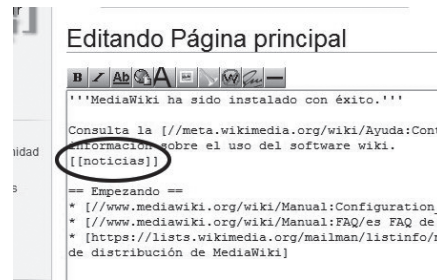


Figura 6.44. Enlace a noticias

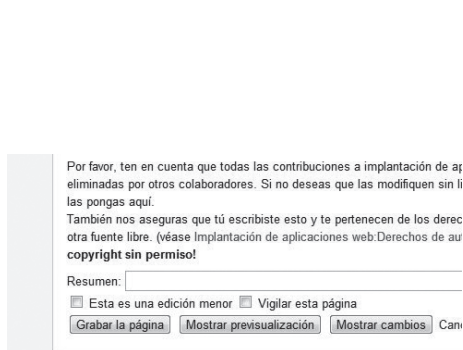


Figura 6.45. Grabar la página



Figura 6.46. Enlace rojo

Al hacer clic en un enlace rojo le llevará a la página de edición del nuevo artículo. Basta con escribir el texto, haga clic en **Grabar la página** y la nueva página será creada.

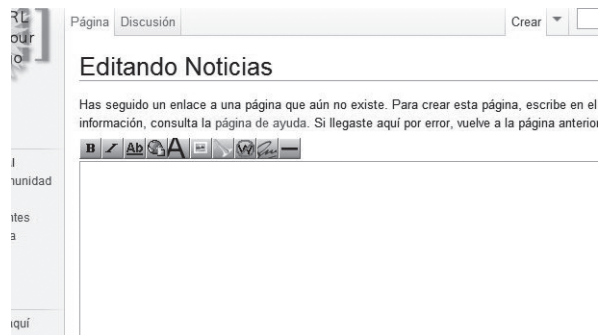


Figura 6.47. Editando la página nueva

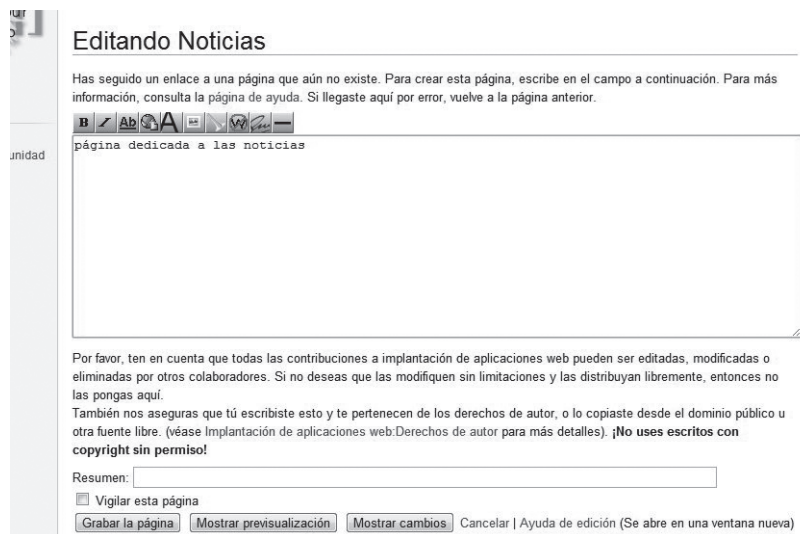


Figura 6.48. Dotar contenido con noticias

Una vez que la página ha sido creada, el enlace cambiará de rojo a azul (color morado para las páginas que ha visitado), indicando que el artículo ya existe.

Página principal

MediaWiki ha sido instalado con éxito.

Consulta la [Guía de usuario](#) para obtener información sobre el uso del software wiki [noticias](#)

Empezando

- [Lista de ajustes de configuración](#)
- [FAQ de MediaWiki](#)
- [Lista de correo de anuncios de distribución de MediaWiki](#)

Figura 6.49. Enlace azul (morado)

Por lo general, ésta es la mejor manera de crear una nueva página, ya que significa que desde el primer momento, la página será enlazada desde, al menos, otro lugar en la wiki. Si va a crear una nueva página sin crear ningún enlace a ella, puede que tenga que preguntarse: ¿realmente esta página encaja con los temas ya incluidos en la wiki? Además, ¿cómo espera que los visitantes encuentren esta página? Normalmente, no crearemos una página nueva sin utilizar un enlace rojo a la misma, ya que no sería accesible mediante enlace desde nuestra wiki.

Otra forma de crear una página es buscar una página que no existe (en el cuadro de búsqueda y botón **Ir** en la parte superior derecha). Tras esto nos mostrará un enlace para crear la nueva página.



Figura 6.50. Buscar

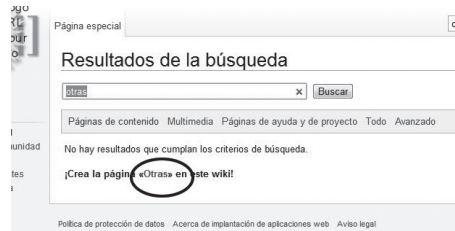


Figura 6.51. Enlace rojo

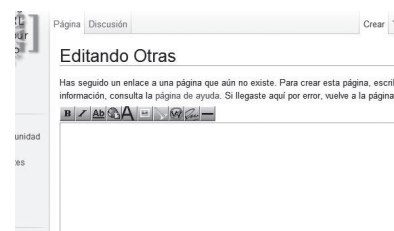


Figura 6.52. Editando

La última forma para crear una página es utilizando directamente la dirección URL de la wiki. La dirección de un artículo de la wiki será `http://miservidor/mediawiki/nuevoarticulo`.

Donde `nuevoarticulo` es el nombre de la página que desea crear. Esto nos llevará a una página en blanco que indica que no existe ningún artículo con ese nombre todavía.

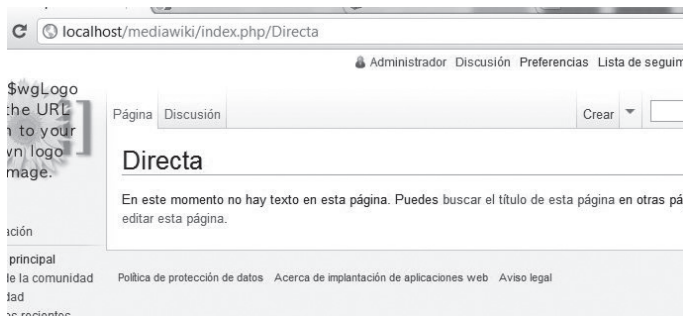


Figura 6.53. Nuevo artículo

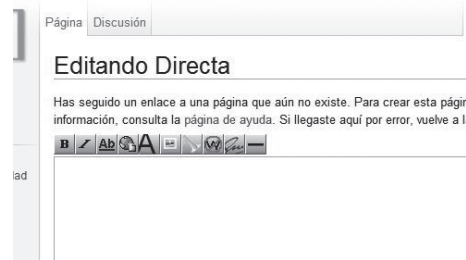


Figura 6.54. Editando la nueva página

No se olvide de crear redirecciones cuando inserte una página nueva, si no nadie podrá acceder al contenido de la misma.

Normalmente, una nueva página wiki puede ser editada por otras personas. Sin embargo, un operador del sistema podría proteger la página para evitar la modificación por usuarios normales.

ACTIVIDADES 6.7



- Cree tres páginas nuevas y redireccionalas a su wiki. Inserte cada una de ellas de una forma distinta. Haga capturas de todo el proceso seguido. No se olvide de la redirecciones.

6.9 PERSONALIZACIÓN DE LA INTERFAZ, PLANTILLAS

A pesar de que la apariencia varía entre las plantillas, en general, todas las plantillas tienen la misma interfaz de componentes. Cada plantilla puede mostrar el componente de la interfaz de manera diferente, mostrando las opciones en diferentes colores o posicionamiento de forma distinta en la página.

Para probar las plantillas disponibles haga clic en **Preferencias** mientras está conectado. A continuación, haga clic en la pestaña **Apariencia**.

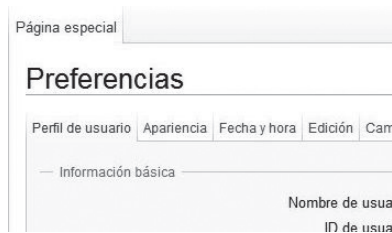


Figura 6.55. Preferencias

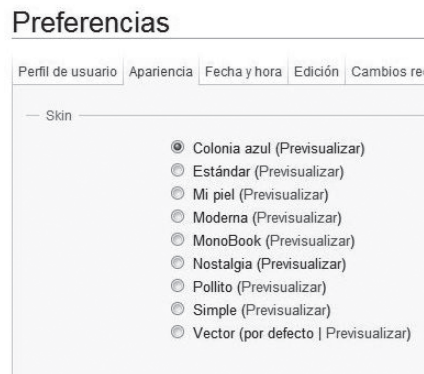


Figura 6.56. Apariencia

Por defecto, MediaWiki incluye los siguientes aspectos: Pollito, Estándar, Colonia Azul, etc. (se pueden apreciar todos en la Figura 6.56). Podemos elegir la plantilla que más nos guste, hay tener en cuenta que algunas plantillas o elementos de diseño que no se mostrarán como se pretende en navegadores de clientes. En términos generales, la plantilla *Vector* (por defecto) o la plantilla *MonoBook* nos asegurará de que se muestre la wiki como se pretende.

Los usuarios familiarizados con Hojas de Estilo en Cascada (CSS) pueden personalizar el archivo de la plantilla actual.

ACTIVIDADES 6.8



- Cambiar la apariencia de su wiki. Hacer capturas de todo el proceso seguido.

6.10 MECANISMOS DE SEGURIDAD INTEGRADOS

En ocasiones, algunos usuarios pueden hacer entradas en una wiki y pueden borrar contenido importante, introducir errores, contenido inapropiado u ofensivo, etc. También pueden incumplir la normativa de la wiki o intentar introducir *spam*. Para evitar esta serie de actuaciones no deseadas podemos llevar a cabo las siguientes acciones:

- ✓ Deshacer rápidamente los cambios en la wiki.
- ✓ Bloquear temporalmente usuarios o direcciones IP, de tal forma que no puedan seguir editando.
- ✓ Si estas actividades se producen siempre en una misma página, podemos proteger esa página.
- ✓ No permitir editar páginas sin estar registrado. También podemos aplicar una variante de esta medida: no poder editar las entradas si no se es un usuario registrado con un mínimo de antigüedad.
- ✓ En casos extremos, generalmente, ataques por medio de herramientas automáticas, podemos bloquear la base de datos de la wiki, no permitiendo así ningún tipo de edición.

Además de estas medidas, en algunas wikis se crean y mantienen toda una serie de herramientas e implementaciones técnicas destinadas a actuar contra el vandalismo: desde foros y canales específicos de IRC hasta programas paralelos de detección y aviso, pasando por usuarios que son en realidad programas *bot*, cuya misión es exclusivamente combatir el vandalismo.

6.11 EDICIÓN

Es muy fácil editar el contenido de una wiki, ya sea la nuestra o alguna en la que queramos colaborar. Es cuestión de unos pocos clics.

- Nos situamos sobre la página que queremos editar.
- Hacemos clic en **Editar**, pestaña que se encuentra en la parte superior derecha de la página.



Figura 6.57. Editar página

- Realizamos los cambios en el texto.

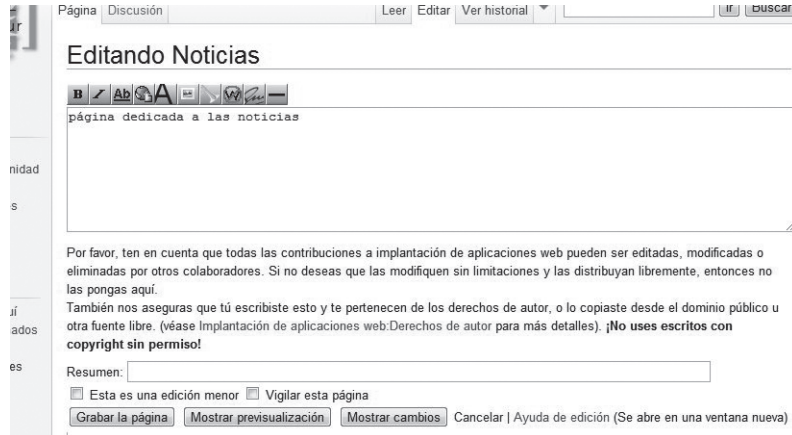


Figura 6.58. Editando la página

- Pulsamos sobre el botón **Grabar la página**.

La regla número uno de la edición de una wiki es ser audaz, investigar y hacer cambios. Otras personas pueden corregir sus errores después, así que tenga confianza y dese una oportunidad. Existen una serie de reglas y políticas que rigen la edición de páginas wiki, pero el “ser valiente” es la regla que domina todas las demás.

En general, hay que tratar de escribir con claridad y concisión, y con el objetivo de hacer algo que mejore los contenidos de la wiki. Una edición puede consistir en párrafos enteros o páginas completas, o puede ser tan simple como la corrección de un error de ortografía.

Cuando sea necesario utilizar algún tipo de formato, por ejemplo, un nuevo título o texto en negrita, se debe hacer utilizando la sintaxis wiki o los botones en la barra de edición anterior.

Antes de guardar un cambio, podemos introducir una pequeña nota en el resumen. No hay que preocuparse en exceso sobre esto, con una pequeña descripción de lo que se acaba de cambiar valdría: “arreglé la ortografía” o “añadí más información acerca de los girasoles”, será más que suficiente.

El resumen, se graba junto con la edición y permite a las personas a seguir los cambios en la wiki con mayor eficacia.

Es una buena idea usar la función **Mostrar previsualización** (ver Figura 6.59) para ver cómo quedarán los cambios antes de guardar la página. Es conveniente revisar nuestro propio trabajo con la vista previa antes de guardar, porque si encontramos un error podremos cambiarlo en lugar de tener que editar después para hacer correcciones.

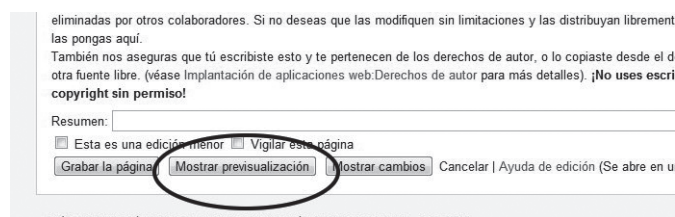


Figura 6.59. Mostrar previsualización

Otra opción es **Mostrar cambios** (ver Figura 6.60), la cual nos permite ver las diferencias entre la versión actual y la versión editada.

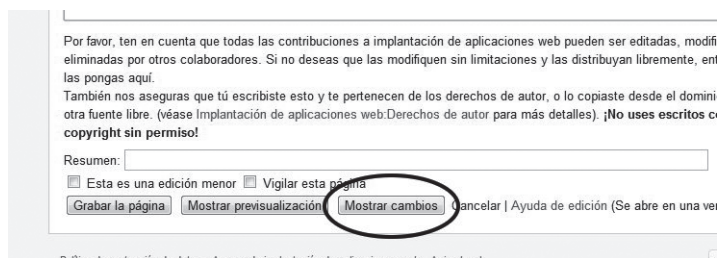


Figura 6.60. Mostrar cambios

ACTIVIDADES 6.9



- Edite la página principal y las tres que hemos creado de nuestra wiki. Haga capturas de todo el proceso seguido.

6.12 COPIAS DE SEGURIDAD

Para hacer una copia de seguridad de todo lo hecho hasta ahora introducimos en el navegador la dirección de nuestro servidor de bases de datos PHPMyAdmin. Nos vamos hasta la pestaña **Exportar**.



Figura 6.61. Exportar

Seleccionamos la base de datos que se creó al instalar la wiki, en nuestro caso llamada **my_wiki**. Marcamos **Guardar a un archivo**.

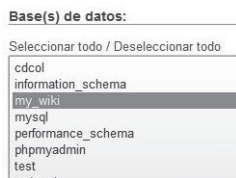
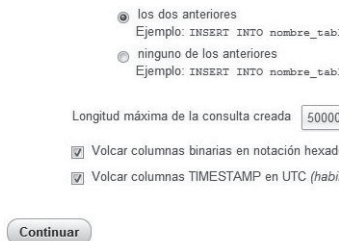


Figura 6.62. Seleccionamos bd

Pulsamos sobre **Continuar**. Debemos guardar el fichero `.sql` que crea pues es la copia de seguridad de la base de datos.



The screenshot shows a web form for a backup process. It has two radio buttons: 'los dos anteriores' (selected) and 'ninguno de los anteriores'. Below each is an example SQL command: 'Ejemplo: INSERT INTO nombre_tab:'. There is a text input field for 'Longitud máxima de la consulta creada' with the value '50000'. At the bottom, there are two checked checkboxes: 'Volcar columnas binarias en notación hexadi' and 'Volcar columnas TIMESTAMP en UTC (habii'. A 'Continuar' button is at the bottom center.

Figura 6.63. Continuar

ACTIVIDADES 6.10



➤ Haga una copia de seguridad de la base de datos de su wiki. Haga capturas de todo el proceso.



RESUMEN DEL CAPÍTULO

Ha llegado el turno de las wikis como objeto de estudio de este capítulo. A lo largo del mismo se introduce el concepto y se presenta la wiki más utilizada en la web, la Wikipedia.

Tras esta introducción, se nos presenta una de las wikis más utilizadas en la actualidad, MediaWiki, ya que es software libre y tiene toda la potencia que puede esperarse de una wiki.

Se ha tratado su instalación, personalización y dotación de contenido a nuestra wiki o a otras wikis que no sean la nuestra, ya que el proceso que deberá seguirse es el mismo que para dotar de contenido a la misma.



EJERCICIOS PROPUESTOS

- 1. ¿Qué es una wiki? ¿Para qué sirve?
- 2. ¿Cuál es la wiki más conocida y utilizada en la web?
- 3. ¿Dónde cree que radica el éxito de la Wikipedia?
- 4. ¿Puede considerarse verdadero y contrastado todo lo que se publica en la Wikipedia? ¿Por qué?
- 5. Plantee algún tema o contenido sobre el que sería interesante realizar una wiki.



TEST DE CONOCIMIENTOS

- 1 El concepto de wiki y las wikis nacen en el año:
 - a) 1993.
 - b) 1995.
 - c) 2000.
 - d) 2001.
 - e) 2011.
 - f) Todas son ciertas.
 - g) Todas son falsas.
- 2 La Wikipedia nace en el año:
 - a) 1993.
 - b) 1995.
 - c) 2000.
 - d) 2001.
 - e) 2011.
 - f) Todas son ciertas.
 - g) Todas son falsas.
- 3 MediaWiki y, en general, las wikis son:
 - a) Gratuitas.
 - b) Gratis las 3 primeras, a partir de la cuarta son de pago.
 - c) De pago.
 - d) De pago solo para empresas, para los usuarios comunes son gratis.
 - e) Todas son falsas.
- 4 Para instalar MediaWiki necesito tener:
 - a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
 - b) Un servidor web con soporte PHP y MySQL.
 - c) Un equipo con Windows.
 - d) Un equipo con Linux.
 - e) Todas son ciertas.
 - f) Todas son falsas.
- 5 Para utilizar MediaWiki o cualquier wiki en general necesito tener:
 - a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
 - b) Un equipo normal con una conexión normal.
 - c) Un equipo con Windows.
 - d) Un equipo con Linux.
 - e) Todas son ciertas.
 - f) Todas son falsas.
- 6 Podemos modificar lo que otra persona ha publicado en la wiki:
 - a) Según el privilegio que tengamos.
 - b) A veces, depende de la wiki.
 - c) Siempre.
 - d) Las respuestas a y b son ciertas.
 - e) Todas son falsas.

7

Moodle

OBJETIVOS DEL CAPÍTULO

- ✓ Definir qué es Moodle.
- ✓ Diferenciar los componentes de Moodle.
- ✓ Dar de alta cursos y conocer los elementos que pueden utilizarse para dotar de contenido a los mismos.
- ✓ Conocer los principales usuarios y papeles desarrollados.

7.1 DEFINICIÓN

Moodle es un Ambiente Educativo Virtual, sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conocen como LMS (*Learning Management System*).

Moodle fue creado por Martin Dougiamas, quien fue administrador de WebCT en la Universidad Tecnológica de Curtin. Basó su diseño en las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el aprendizaje colaborativo. Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que le ayuda a construir ese conocimiento con base en sus habilidades y conocimientos propios, en lugar de simplemente publicar y transmitir la información, que se considera que los estudiantes deben conocer.

La primera versión de la herramienta apareció el 13 de marzo de 1987. A partir de ahí han aparecido nuevas versiones de forma regular. Hasta julio de 2008, la base de usuarios registrados incluye más 21 millones, distribuidos en 46.000 sitios en todo el mundo y está traducido a, más o menos, 91 idiomas.

La palabra MOODLE son las iniciales de *Module Object-Oriented Dynamic Learning Environment* (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos).

En términos de arquitectura, Moodle es una aplicación web que se ejecuta sin modificaciones en Unix, GNU/Linux, OpenSolaris, FreeBSD, Windows, Mac OS X, NetWare y otros sistemas que soportan PHP, incluyendo la mayoría de proveedores de alojamiento web.

Los datos se almacenan en una sola base de datos SQL. En Moodle 1.6 las únicas opciones eran MySQL y PostgreSQL. Desde la versión 1.7, publicada en noviembre de 2006, los instaladores pueden elegir entre diversos motores de bases de datos (donde se incluyen los soportados por la versión 1.6, además de Oracle y Microsoft SQL Server).

7.2 LICENCIAS DE USO

El paquete de software Moodle está distribuido bajo los términos de la licencia GPL. Este programa es software libre; puede ser redistribuido y/o modificado bajo los términos de la Licencia Pública General GNU (*GNU General Public License* o GPL).

La documentación de Moodle se proporciona bajo los mismos términos de la GPL que el software Moodle.

Moodle y la Documentación de Moodle son distribuidos con la esperanza de que sean útiles, pero **sin ninguna garantía**, es decir, el creador no garantiza el producto frente a fallos o problemas que pudieran derivarse del uso del mismo.

7.3 REQUERIMIENTOS DE FUNCIONAMIENTO

Moodle está desarrollado principalmente en GNU/Linux usando Apache, MySQL y PHP, aunque funciona con PostgreSQL y, en los sistemas operativos Windows (a partir del XP), MacOS X y Netware 6.

Los requerimientos de Moodle son los siguientes:

- **Un servidor web.** La mayoría de los usuarios usan Apache, pero Moodle debe funcionar bien en cualquier servidor web que soporte PHP, como el IIS (*Internet Information Server*) de las plataformas Windows.
- **Una instalación de PHP en funcionamiento** (versión 4.3.0 o posterior). PHP 5 está soportado a partir de Moodle 1.4. (tenga cuidado con PHP-Accelerator ya que se han detectado problemas con él). Moodle 2 necesita como mínimo la versión 5.2.8. Si se usa PHP 5.3 hay que asegurarse que la versión es superior a la 5.3.2.
- **Una base de datos.** MySQL o PostgreSQL, que están completamente soportadas y recomendadas para su uso con Moodle.

La mayoría de los servicios de alojamiento web (*hosting*) soportan todo esto por defecto. Si ha contratado los servicios de alguno de los pocos de alojamiento web que no soportan estas características, pregúnteles por qué no lo hacen y considere la posibilidad de trasladar su sistema a otro sitio.

7.4 INSTALACIÓN

Existen dos formas de obtener Moodle: como un paquete comprimido y a través de CVS. Tras descargar y descomprimir el archivo, o actualizar los archivos vía CVS, tendrá un directorio llamado **moodle** que contiene varios archivos y carpetas.

Puede colocar la carpeta completa en el directorio de documentos de su servidor web, en cuyo caso el sitio estará localizado en: `http://servidor/moodle`

Moodle detectará la configuración necesaria y le guiará a través de algunas pantallas para ayudarle a crear el archivo de configuración llamado `config.php`. Al final del proceso, Moodle intentará escribir el archivo en el lugar apropiado, pero si esto no fuera posible puede presionar un enlace que nos aparece para descargarlo en nuestro ordenador desde el instalador y después subirlo al directorio principal de Moodle en el servidor.

Al mismo tiempo, el instalador comprobará las características de su servidor y le sugerirá cómo resolver cualquier problema.

PRÁCTICA 7.1



➔ Descarga de Moodle.

Nos dirigimos a la página web de Moodle: http://moodle.org/?lang=es_es. Dentro de esta página web nos vamos a la sección **Descargas** (Figura 7.2).

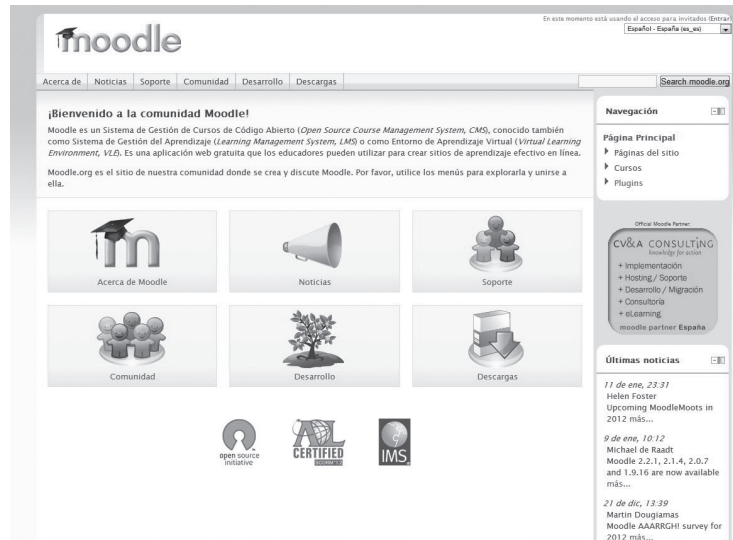


Figura 7.1. Web de Moodle



Figura 7.2. Descargas

Descargaremos el paquete estándar de Moodle (Figura 7.3). Dentro de la última versión estable, podemos elegir en formato *.tgz* o *.zip* (dependiendo del sistema operativo donde vayamos a instalarlo). En nuestro caso hemos elegido *.zip* pues lo vamos a instalar en Windows (Figura 7.4).

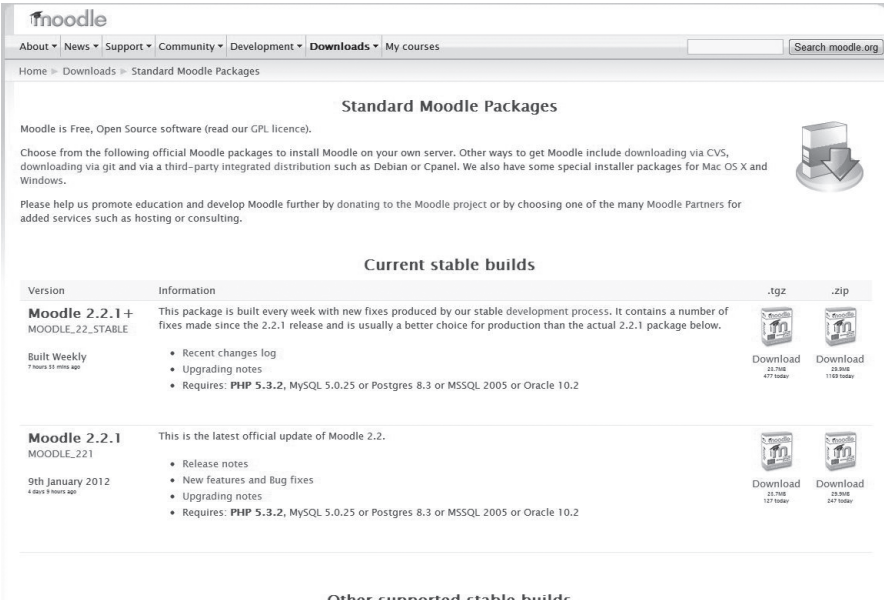


Figura 7.3. Standard Module Packages

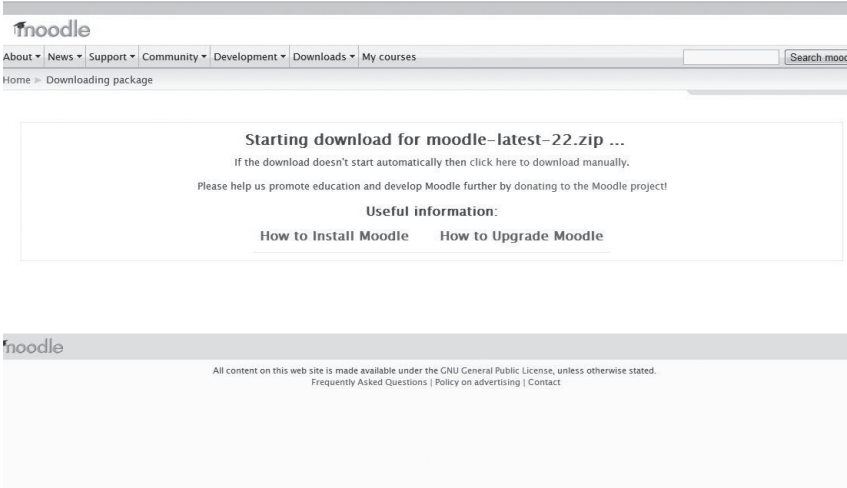
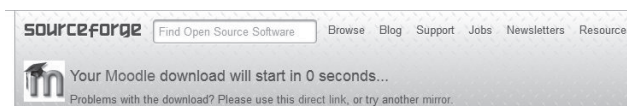


Figura 7.4. Starting download

Esperamos que aparezca el diálogo de descarga y descargamos el archivo. Una vez descargado, descomprimos el contenido del archivo descargado en la raíz de nuestro servidor web, en nuestro caso la carpeta *htdocs* de *xampp*.



Medir ancho de banda

Programa de

Figura 7.5. Descarga

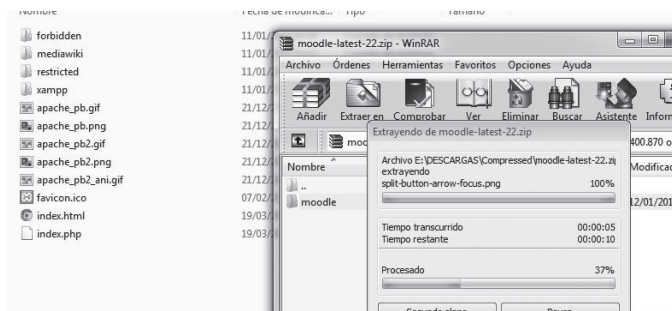


Figura 7.6. Descomprimir en htdocs

ACTIVIDADES 7.1



- Descargue, descomprima, en una carpeta llamada *moodle*, y copie en su servidor la última versión de Moodle que encuentre en la Web. Realice capturas de todo el proceso.

PRÁCTICA 7.2



- Instalación de Moodle en nuestro servidor. Partimos de: <http://miservidor/moodle>

En primer lugar debemos elegir el idioma, tanto de instalación como de uso de nuestro sitio Moodle (Figura 7.7). En la siguiente pantalla debemos confirmar los directorios de instalación, tanto la dirección web a utilizar como el directorio de Moodle y el de datos.

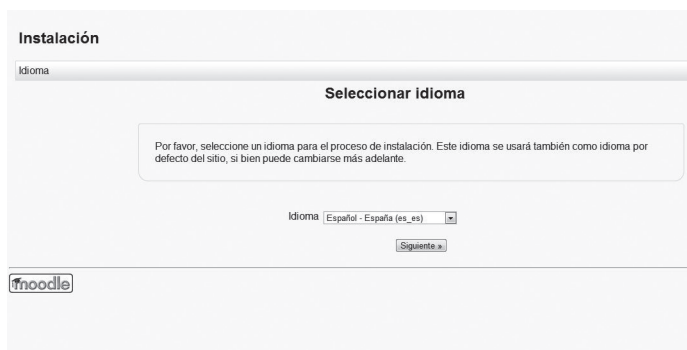


Figura 7.7. Seleccionar idioma

The screenshot shows the 'Instalación' (Installation) screen with the 'Rutas' (Routes) tab selected. The title is 'Confirme las rutas' (Confirm the routes). It contains three sections: 'Dirección Web' (Web Address) with a text box containing 'http://localhost/moodle'; 'Directorio Moodle' (Moodle Directory) with a text box containing 'E:\xampp\htdocs\moodle'; and 'Directorio de Datos' (Data Directory) with a text box containing 'E:\xampp\moodledata'. At the bottom, there are two buttons: '< Anterior' and 'Siguiente >'. The Moodle logo is in the bottom left corner.

Figura 7.8. Confirme las rutas

En la siguiente pantalla debemos **seleccionar el controlador de la base de datos**, por defecto el sistema utilizará MySQLi (que es un MySQL mejorado) (Figura 7.9). Después de hacer clic en **Siguiente** nos aparecerán los **ajustes de la base de datos**. Aquí debemos introducir la dirección del servidor, nombre usuario, contraseña y demás datos para configurar nuestra base de datos (Figura 7.10).

The screenshot shows the 'Instalación' (Installation) screen with the 'Base de datos' (Database) tab selected. The title is 'Seleccione el controlador de la base de datos' (Select the database controller). It contains a text box with the message: 'Moodle soporta varios tipos de servidores de base de datos. Por favor, póngase en contacto con el administrador del servidor si no sabe qué tipo usar.' Below this is a dropdown menu labeled 'Tipo' (Type) with 'MySQL mejorado (native/mysqli)' selected. At the bottom, there are two buttons: '< Anterior' and 'Siguiente >'. The Moodle logo is in the bottom left corner.

Figura 7.9. Controlador de la BD

The screenshot shows the 'Instalación' (Installation) screen with the 'Base de datos' (Database) tab selected. The title is 'Ajustes de base de datos' (Database settings). It contains a section for 'MySQL mejorado (native/mysqli)' with the following text: 'Ahora tiene que configurar la base de datos donde se almacenarán la mayoría de los datos de Moodle. La base de datos solo podrá crearse si el usuario de la base de datos tiene los permisos necesarios. El nombre de usuario y la contraseña ya deben existir. El prefijo de la tabla es opcional.' Below this are five text boxes: 'Servidor de la base de datos' (localhost), 'Nombre de la base de datos' (moodle), 'Usuario de la base de datos' (root), 'Contraseña de la base de datos' (password), and 'Prefijo de tablas' (mdl_). At the bottom, there are two buttons: '< Anterior' and 'Siguiente >'. The Moodle logo is in the bottom left corner.

Figura 7.10. Ajustes del servidor BD

Tras esto nos aparecerá la pantalla con la licencia de Moodle. Una vez hemos leído y comprendido los términos de la licencia pulsamos sobre **Continuar** (Figura 7.11.). En la siguiente pantalla se realizarán las **comprobaciones del servidor** donde vamos a instalar Moodle (Figura 7.12).



Figura 7.11. Licencia



Figura 7.12. Comprobación del servidor

Si el paso anterior se ha efecutado con éxito se inicia la instalación de archivos de Moodle en nuestro servidor. Este proceso puede tardar un tiempo, tenga paciencia (Figuras 7.13 y 7.14).

Sistema
Éxito
qtype_calculated
Éxito
qtype_calculatedmulti
Éxito
qtype_calculatedsimple
Éxito
qtype_description
Éxito
qtype_essay
Éxito

Figura 7.13. Instalación de archivos

scormreport_interactions
Éxito
workshopform_accumulative
Éxito
workshopform_comments
Éxito
workshopform_numerrors
Éxito
workshopform_rubric
Éxito
workshopallocation_manual
Éxito
workshopallocation_random
Éxito
workshopeval_best
Éxito
Continuar

Figura 7.14. Instalación de archivos

Llegamos a la página de creación del usuario administrador del sitio moodle, en la que debemos rellenar una serie de datos relacionados con este usuario (Figura 7.15 y 7.16).

Instalación

En esta página debería configurar su cuenta de administrador principal, que le dar absoluto sobre el sitio. Asegúrese de que usa un nombre de usuario y contraseña una dirección de correo electrónico válida. Más adelante podrá crear más cuentas

General

Nombre de usuario*
admin

Escoger un método de identificación: ①
Cuentas manuales

La contraseña debería tener al menos 8 caracte(es), al menos 1 dígito(s), al menos 1 minúscula(s) alfanuméricos

Nueva contraseña* ①
***** ☒ Desenmascarar

Forzar cambio de contraseña ☐

Nombre*
Admin

Apellido*
Usuario

Dirección de correo*
implantaciondeaplicacionesweb@gni

Mostrar correo
Mostrar a todos mi dirección de correo

Formato de correo
Formato HTML

Figura 7.15. Cuenta administrador

Tipo de resumen de correo
Sin resumen (un correo por cada mensaje del foro)

Subscripción automática al foro
Si, cuando envíe un mensaje suscríbame a ese foro

Cuando edite texto
Usar el editor de HTML

AJAX y Javascript
Si: usar características web avanzadas

Lector de pantalla ①
No

Ciudad*

Seleccione su país*
Seleccione su país...

Zona horaria
Hora local del servidor

Idioma preferido
Español - España (es_es)

Descripción ①

Figura 7.16. Cuenta administrador

Una vez hemos completado los datos del administrador, debemos rellenar los **ajustes de la página principal** del sitio (Figura 7.17). Una vez hayamos insertado estos datos y pulsado sobre **Guardar cambios** nos aparecerá una pantalla donde podemos crear cursos nuevos para nuestro gestor (Figura 7.18).

Nuevos ajustes - Ajustes de la página principal	
Nombre completo del sitio fullname	Implantación de Aplicaciones Web
Nombre corto para el sitio (una palabra) shortname	IAW
Descripción de la página principal summary	Fuente Tamaño Párrafo Imágenes Tablas Listas Links Fuente de color Fondo de color Fuente de fondo de color B I U ABC x² BB UL OL Link Unlink Image Table Row Col Merge Split Undo Redo Print Help Teleformación para Implantacion de Aplicaciones Web Ruta: p Esta descripción del sitio aparecerá en la página principal
Nuevos ajustes - Gestionar identificación	
Registrarse a sí mismo registerauth	Deshabilitar Valor por defecto: Deshabilitar Escoja qué conector ("plugin") de identificación manejar cuando los usuarios se registren a sí mismos.

Figura 7.17. Instalación de Moodle

Implantación de Aplicaciones Web

Navegación

- Página Principal
 - Área personal
- ▶ Páginas del sitio
- ▶ Mi perfil

Cursos disponibles

No hay cursos en esta categoría

[Agregar un nuevo curso](#)

Teleformación
Implantacion
Web

Calendario

<	ener	
Dom	Lun	Már
1	2	3
2	9	10
15	16	17
22	23	24
29	30	31

Ajustes

- ▼ Ajustes de la página principal
 - ✎ Activar edición
 - 📅 Editar ajustes
 - ▶ Usuarios
 - 🔒 Copia de seguridad
 - 🔄 Restaurar
 - ▶ Banco de preguntas
- ▶ Ajustes de mi perfil
- ▶ Administración del sitio

Buscar

Usted se ha identificado como Admin Usuario (Salir)

Figura 7.18. Instalación de Moodle

Si pulsamos sobre el botón **Agregar un curso nuevo** nos aparecerá una nueva pantalla donde tendremos que rellenar los datos oportunos del nuevo curso (Figura 7.19). Una vez completados estos datos, ya tenemos nuestro curso creado. Automáticamente nos llevará dentro del curso a la sección usuarios, para que procedamos al alta de usuarios en este nuevo curso (Figura 7.20).

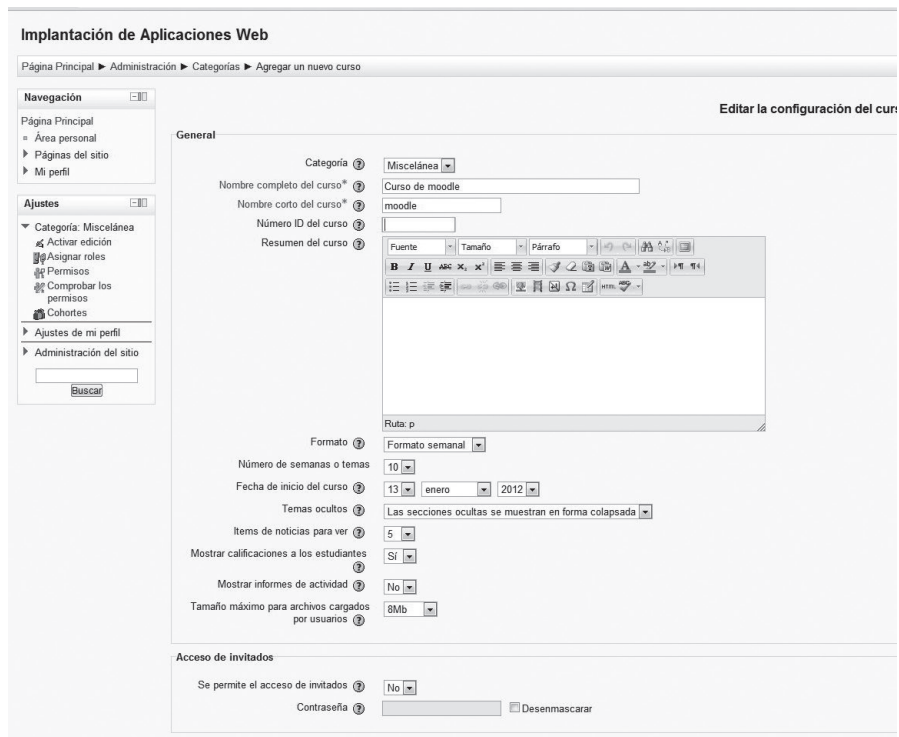


Figura 7.19. Instalación de Moodle

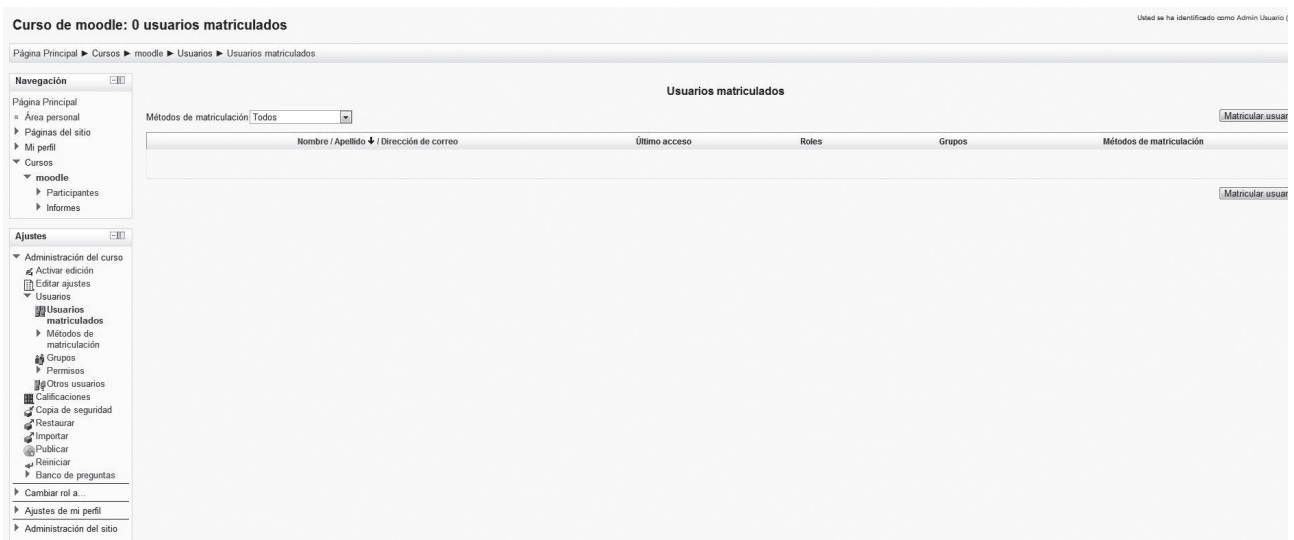


Figura 7.20. Instalación de Moodle

En nuestro caso, como solo disponemos de un usuario (el administrador), lo matriculamos con privilegios de administrador (valga la redundancia) del curso recién creado (Figura 7.21). Finalizamos la matriculación de alumnos, después podremos matricular a los usuarios que queramos y entraremos dentro del curso creado anteriormente (Figura 7.22).

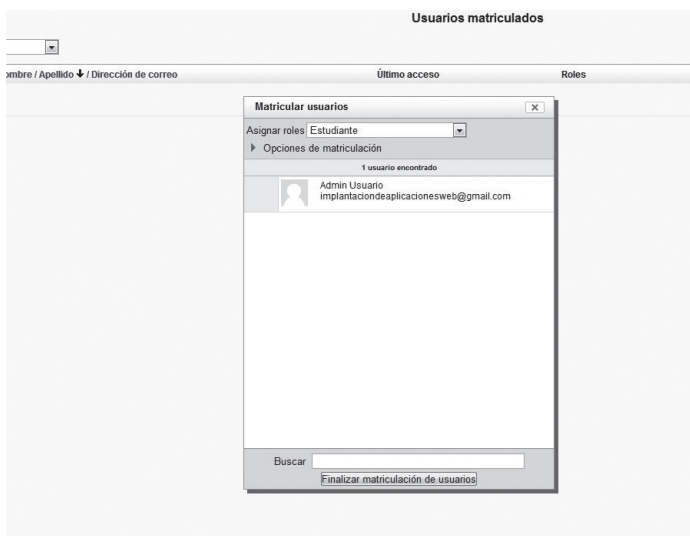


Figura 7.21. Instalación de Moodle

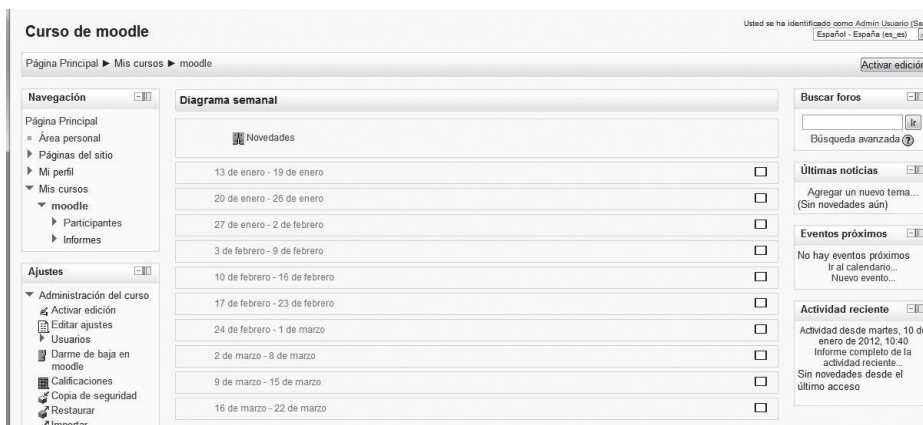


Figura 7.22. Instalación de Moodle

ACTIVIDADES 7.2



- Instale en su servidor web la versión de Moodle que ha descargado en la Práctica 7.1. Durante la instalación, cuando pida el nombre del sitio, ponerle el nombre del alumno/a. Haga capturas de todo el proceso.

7.5 CÓMO CREAR UN CURSO

- Para poder crear un curso debemos estar identificados como un usuario con condición de creador de cursos o administrador, ir a la página principal del sitio.
- Hacer clic en **Cursos** desde el menú **Administración del sitio** (Figura 7.23).
- Hacer clic en el botón **Agregar/editar cursos** y, a continuación, **Agregar un nuevo curso** (Figuras 7.23 y 7.24).

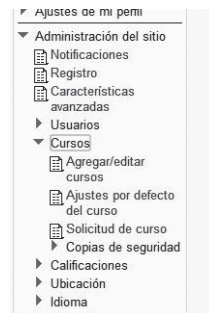


Figura 7.23. Administración del sitio, cursos



Figura 7.24. Agregar un nuevo curso

- Nos mostrará un formulario bastante extenso. Rellene los diferentes campos y use los botones de ayuda si los necesita. Recuerde que siempre puede volver a editarlo y que las opciones que vienen por defecto, suelen servir para la mayoría de los casos (Figura 7.25).

Figura 7.25. Nuevo curso

- Pulsar en **Guardar cambios** y aparecerá el nuevo curso en la página del sitio (Figura 7.26).

Figura 7.26. Guardar cambios

Sugerencias:

- Debe conocer la opción de grupos si desea crear varios cursos con contenidos semejantes, pero con grupos de alumnos diferentes. La opción de grupos hace más fácil mantener un único curso actualizado.
- El nombre corto del curso se envía en el asunto de cada mensaje automático y debe ser identificativo si los alumnos suelen estar inscritos en más de uno.

ACTIVIDADES 7.3



- Cree un curso llamado *Implantación de Aplicaciones Web*. Haga capturas de todo el proceso.

7.6 ESTRUCTURA

Los cursos en Moodle se estructuran mediante un bloque central, que suele contener la lista de temas del curso (o una lista de semanas) y dos columnas laterales que contienen otros bloques opcionales. Para cada tema o semana se especifican una serie de actividades y recursos. Los tipos disponibles son los siguientes:

7.6.1 FOROS

Los foros en Moodle tienen una importancia particular, ya que su filosofía se basa en la teoría del constructivismo social y es ahí donde se desarrolla ese concepto, en el seno Moodle. Para ello y por ello, se tienen varios tipos de foros, cada uno con un objetivo bien definido. Así tenemos:

- **Foros de debate sencillo:** simplemente un intercambio de ideas sobre un solo tema, todo en una página, respondiendo a un único planteamiento inicial. Útil para debates cortos y muy concretos.
- **Foro Normal, para uso general:** foro abierto donde cualquiera puede empezar un nuevo tema de debate cuando quiera. Este es el foro más adecuado para uso general.
- **Foro Un Debate por Persona:** cada persona puede plantear un nuevo tema de debate (y todos pueden responder). Esta modalidad es útil cuando quiera que cada alumno o miembro del grupo empiece una discusión sobre, digamos, sus conclusiones sobre el tema de la semana y que todos los demás le respondan.



Novedades y anuncios				
Agregar un nuevo tema				
Tema	Comenzado por	Réplicas	Último mensaje	
CONSULTAS EN MYSQL	 Juan Ferrer Martinez	0	Juan Ferrer Martinez mié, 18 de ene de 2012, 09:53	
Añadida Práctica 20	 Juan Ferrer Martinez	0	Juan Ferrer Martinez mar, 13 de dic de 2011, 20:34	

Figura 7.27. Foros

En cualquiera de los casos, el enviar nuevos temas o respuestas podría estar restringido por la configuración del foro, aunque no es lo común. Al acceder a cualquier foro, si ya han sido creados temas, podrá ver y revisar incluso las posibles respuestas que ya se hayan añadido, justo debajo de la descripción del foro, en la parte superior de la página.

Si el foro es de tipo **Normal** o del tipo **Un Debate por Persona** y aún no ha sido inaugurado, encontrará el enlace **Añadir un nuevo tema de debate**, mediante el cual podrá agregar un nuevo mensaje en el foro.

Si el foro ya tiene temas abiertos y entra en alguno de ellos, podrá ver el enlace **Responder** dentro de las entradas de los que ya han participado en el foro (incluyendo las suyas).

Ambos enlaces, al hacer clic sobre ellos, le llevarán a una nueva pantalla donde se le pedirá que introduzca un título (automático si está respondiendo a un tema) y el mensaje. Así mismo, debajo del mensaje que piensa agregar al foro, encontrará una serie de botones que le ayudarán a configurar su mensaje.

- **Suscripción:** desde aquí indicará al sistema si desea o no hacerle seguimiento a esa discusión. Hacerle seguimiento quiere decir recibir un correo cada vez que un mensaje sea agregado a ese tema. Esto puede ser de gran utilidad, pues si planteamos alguna cuestión en el foro no tendremos que asomarnos periódicamente para comprobar si hemos sido respondidos, pues recibiremos un mensaje de correo electrónico en el momento en el que se produzca esta circunstancia.

Figura 7.28. Suscripción

- **Archivo adjunto:** opcionalmente tiene la posibilidad de subir al servidor un archivo local de su disco duro, adjuntado a su mensaje para que los demás puedan verlo. Solo debe presionar el botón **Examinar**, buscarlo en su disco duro y listo.

A continuación de que haya escrito y configurado su mensaje solo debe hacer clic sobre el botón **Enviar al Foro** (ver Figura 7.28), con lo que el mensaje será agregado como respuesta o como tema del foro.

A partir de ese momento, tendrá un período de tiempo determinado configurado por el administrador de Moodle para hacer cualquier cambio a su mensaje. Pasado este tiempo no podrá modificarlo y el sistema enviará un correo a todos los estudiantes que están haciéndole seguimiento al tema en cuestión.

De igual forma, en cualquier momento puede eliminar sus propias entradas al foro haciendo clic sobre el botón **Borrar** que aparece en todos los mensajes que haya colocado.

Figura 7.29. Editar, borrar entrada en el foro

Los foros son evaluables, en efecto, dependiendo de la configuración del foro, los mensajes que se dejen en él podrían estar siendo evaluados por el tutor del curso. Así mismo, según la configuración, el alumno podría o no ver sus notas o las calificaciones de los demás. La escala de evaluación también puede variar de un curso a otro.

ACTIVIDADES 7.4



- Acceda al foro novedades de su curso creado en la práctica anterior y deje un mensaje en dicho foro. Haga capturas de todo el proceso.

7.6.2 DIARIOS

Este módulo es muy importante para la actividad reflexiva. El profesor propone a los alumnos reflexionar sobre diferentes temas y los estudiantes pueden responder y modificar sus respuestas a través del tiempo. La respuesta es privada y solo puede ser vista por el profesor, quien puede responder y calificar cada vez que quiera.

7.6.3 APUNTES, MATERIALES O RECURSOS

Los recursos son elementos que contienen información que puede ser leída, vista, bajada de la Red o usada de alguna forma para extraer información de ella.

Las actividades son elementos que le piden que haga algún trabajo basado en los recursos que ha utilizado.

Algunos recursos son:

- Páginas de texto plano.
- Fragmentos HTML.
- Archivos cargados en el servidor.
- Enlaces web.
- Páginas web.

ACTIVIDADES 7.5



- Añada tres apuntes a su curso. El primero será un documento PDF que subiremos a la plataforma, el segundo será un enlace web y, el tercero, será una página de texto. Haga capturas de todo el proceso.

7.6.4 TAREAS

Las tareas permiten a los profesores asignar actividades a los estudiantes, que consisten en preparar contenidos digitales (de cualquier tipo) que el alumno podrá subir al curso. Las tareas típicas son ensayos, monografías, redacciones, etc. El profesor puede calificar estas tareas y usarlo como parte de la calificación del curso.

Un profesor/a puede, mediante Moodle, solicitar a un alumno/a la realización de una tarea que tendrá que desarrollar. El resultado de la misma, normalmente, consistirá en la creación de un archivo en formato digital (documento de procesador, imagen, sonido, etc.) que el alumno/a podrá subir al curso mediante un formulario para su corrección.

Ver 3 tareas enviadas

Elimina la base de datos con tu nombre y envía una captura del proceso

Disponible desde:	miércoles, 5 de octubre de 2011, 10:10
Fecha de entrega:	jueves, 20 de octubre de 2011, 10:10

Subir un archivo

Figura 7.30. Tareas

También es posible proponer a los alumnos/as una tarea que no implique la subida de un archivo, pero que sí lleve asociada una calificación.

Se ha incorporado la posibilidad de que el trabajo que tenga que escribir el alumno lo haga en una ventana que se habilite al efecto.

En todos los casos el profesor o tutor podrá, además de calificar la tarea, realizar algún comentario al respecto.

Una vez enviado el trabajo, aparece una pantalla como la de la Figura 7.31, en la cual puede reenviar la tarea sustituyendo la versión anterior, siempre y cuando se haya configurado para ello. Una vez que el profesor haya evaluado la tarea, podrá ver la calificación, como puede verse en la Figura 7.32.

Figura 7.31. Envío de tarea

Figura 7.32. Calificación

ACTIVIDADES 7.6



- Añada dos tareas a su curso. La primera consistirá en subir un solo archivo a la plataforma, la segunda será una tarea de texto en línea.

7.6.5 CUESTIONARIOS

Este módulo permite que el profesor diseñe y plantee cuestionarios. Estos cuestionarios pueden ser: opción múltiple, falso/verdadero y respuestas cortas. Los cuestionarios pueden permitir múltiples intentos. Cada intento se corrige automáticamente y cuando vuelva a entrar en el cuestionario verá que han ido cambiando sus resultados en los sucesivos intentos. El profesor puede decidir si mostrar la calificación y/o las respuestas correctas a los alumnos una vez concluido el cuestionario. Además, tiene la posibilidad de usarse como parte de la calificación del curso.

Una vez pulsado sobre el enlace correspondiente, el sistema nos mostrará la siguiente pantalla (Figura 7.33):

Figura 7.33. Cuestionarios

Si existieran intentos anteriores, estos figuran en la pantalla con el tiempo consumido y la fecha de realización.

Para comenzar a responder el cuestionario debemos presionar el botón **Comenzar**. Los tipos de preguntas que se pueden presentar en un cuestionario son las siguientes:

- **Opción múltiple:** Moodle ofrece una gran flexibilidad al crear este tipo de pregunta común. Puede crear preguntas de respuesta simple y múltiple, presentar imágenes en la pregunta y ponderar respuestas individuales. Hay dos tipos de preguntas de elección múltiple: de una respuesta y de múltiples respuestas.
- **Preguntas de respuesta única:** estas preguntas permiten una y solo una respuesta proporcionando botones de elección al lado de las respuestas. Detallará puntuaciones no-negativas para cada respuesta, generalmente una puntuación de cero para las respuestas erróneas, puntuaciones máximas para las respuestas correctas y puntuaciones parciales para respuestas parcialmente correctas.
- **Preguntas de respuesta múltiple:** estas preguntas permiten seleccionar una o más respuestas por medio de casillas de verificación. Cada contestación puede tener una puntuación positiva o negativa, lo que implica que seleccionar TODAS las opciones no necesariamente supone una buena puntuación. Si la puntuación total es negativa, entonces la puntuación resultante para esta pregunta será cero. *Atención:* es posible crear preguntas que tengan una puntuación mayor que el 100%.
- **Respuesta corta:** en respuesta a una pregunta (la cual puede incluir una imagen) el alumno escribe una palabra o frase corta. Puede haber varias respuestas correctas posibles, cada una con una puntuación diferente. Si selecciona la opción **Sensible a las mayúsculas**, podrá tener puntuaciones diferentes para “Palabra” o “palabra”. Las respuestas se comparan exactamente, así que hay que tener cuidado con la ortografía.
- **Númerica:** desde el punto de vista de un alumno, una pregunta numérica es muy semejante a una pregunta de respuesta corta. La diferencia es que la pregunta numérica es un número y permite un margen de error. Esto permite habilitar un rango continuo de posibles respuestas acertadas. Por ejemplo, si la respuesta es 30 con un error de 5, entonces cualquier número entre 25 y 35 se aceptará como correcto.
- **Emparejamiento:** después de una introducción (opcional), se presenta una lista de preguntas junto con una lista de respuestas. El alumno debe seleccionar la respuesta correcta que corresponda a cada pregunta de la lista. El alumno debe seleccionar una respuesta que corresponda en cada subpregunta. Cada subpregunta es ponderada por igual para conseguir la puntuación total de la pregunta completa. *Nota:* las preguntas de emparejamiento no permiten comentarios (*feedback*).
- **Preguntas incrustadas (Cloze):** las preguntas incrustadas (*Cloze*) consisten en un texto que puede integrar (en formato Moodle) varias preguntas incrustadas de tipo opción múltiple, preguntas cortas y numéricas. Actualmente no disponemos de una interfaz gráfica para crear este tipo de preguntas, de modo que es preciso especificar el formato de la pregunta usando la caja de texto o bien importarla desde archivos externos.
- **Preguntas tipo ensayo:** en contestación a una pregunta (que puede incluir una imagen) el alumno escribe una respuesta en formato ensayo. Pueden editarse tres campos cuando se crea una pregunta de ensayo: el título de la pregunta, el cuerpo de la pregunta y el comentario (*feedback*), que puede mostrarse en un momento elegido por el facilitador. Las preguntas tipo ensayo no son calificadas en tanto no hayan sido revisadas por un profesor o facilitador por medio del uso de la opción **Calificación Manual**. Cuando se califica manualmente una pregunta de ensayo, el calificador podrá escribir un comentario personalizado en respuesta al ensayo del alumno y podrá así mismo asignar una puntuación al ensayo.
- **Aleatoria:** nos permite seleccionar un número de preguntas que serán aleatoriamente seleccionadas cada vez que un estudiante intenta responder el examen.

- **Emparejamiento de respuesta corta aleatoria:** desde la perspectiva del alumno, equivale a una pregunta de emparejamiento. La diferencia es que las subpreguntas se extraen aleatoriamente del conjunto de preguntas de respuesta corta en la categoría actual. Después de una introducción opcional, se le presentan al alumno varias subpreguntas y varias respuestas mezcladas. Hay una respuesta correcta para cada pregunta. El alumno debe seleccionar una respuesta que corresponda con cada subpregunta. Cada subpregunta es ponderada por igual para conseguir la puntuación total de la pregunta completa. Las preguntas y respuestas se extraen aleatoriamente de la pila de **Preguntas Cortas**, en la categoría **Actual**. Cada intento de resolver un cuestionario tendrá diferentes preguntas y respuestas.
- **Descripción:** este formato no es una pregunta en sentido estricto. Se limita a mostrar un texto (y si lo desea algún gráfico) sin necesidad de responder. Puede ser útil, por ejemplo, para dar información previa antes de un grupo de preguntas.
- **Calculadas:** las preguntas calculadas ofrecen una forma de crear preguntas numéricas individuales por medio del uso de tarjetas que son sustituidas por los valores individuales cuando se responde el cuestionario.
- **Tipos de preguntas no estándar:** además de los tipos de preguntas, ya descritos, que forman parte de la distribución esencial de Moodle, se pueden añadir como *plugins* otros tipos preguntas desarrolladas por la comunidad.
- **Rendered Matching:** rellene los Blancos; es el popular formato donde se le presenta al estudiante un texto donde faltan algunas palabras que debe llenar. En Moodle, dichos “espacios en blanco” pueden requerir que introduzca una respuesta corta, una numérica o que realice una selección múltiple.

ACTIVIDADES 7.7



- Añada un cuestionario a su curso, como mínimo el cuestionario deberá tener cinco preguntas y ser de tipos diferentes (numéricas, respuesta corta, etc.). Haga capturas de todo el proceso.

7.6.6 CONSULTAS

Las consultas son muy sencillas: el profesor hace una pregunta y determina ciertas opciones, de las cuales los alumnos elegirán una. Es útil para conocer rápidamente el sentimiento del grupo sobre algún tema, para permitir algún tipo de elecciones del grupo o para efectos de investigación.

Para participar haga clic sobre la actividad y será llevado a una página de consulta. En la parte superior tiene la descripción colocada por el profesor mientras que en la parte inferior tiene las opciones que puede elegir.

Figura 7.34. Consulta de ejemplo

Simplemente seleccione la opción que crea pertinente y presione el botón **Guardar mi elección**, con lo que su voto será guardado. La próxima vez que entre, dependiendo de la configuración de la consulta, es posible que vea un gráfico de barras mostrando las votaciones hasta ese momento (con la posibilidad de ver a quién pertenece cada voto también según configuración) o bien podrá ver cuál fue su respuesta.

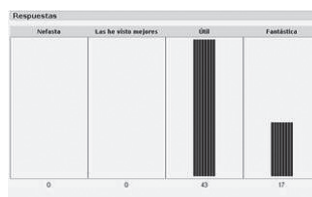


Figura 7.35. Resultado de la consulta

ACTIVIDADES 7.8



➤ Añada una consulta a su curso de como mínimo cuatro posibilidades. Haga capturas de todo el proceso.

7.6.7 ENCUESTAS

El módulo de encuestas provee una serie de instrumentos probados para estimular el aprendizaje en Internet. Los profesores pueden utilizar este módulo para aprender sobre sus alumnos y reflexionar sobre su práctica educativa.

Moodle tiene 2 tipos de encuestas las cuales se enumeran a continuación. Están incluidas por ser particularmente útiles para la evaluación de la educación en línea, así como para identificar ciertas tendencias que pueden estar dándose entre sus participantes.

COLLES - Constructivist On-Line Learning Environment Survey (Encuesta sobre Ambiente Constructivista Educativo en Línea).

El COLLES comprende 24 premisas agrupadas en seis escalas, cada una de las cuales nos ayuda a formular una pregunta clave sobre la calidad del ambiente educativo en línea. Esta encuesta se puede realizar desde distintos puntos de vista: Ambiente Ideal, Ambiente Real y Ambiente Ideal vs. Real. En todos, las escalas que se determinan son:

- **Relevancia:** ¿Cómo de importante es la educación en línea para la práctica profesional de los estudiantes?
- **Reflexión:** ¿La educación en línea estimula el pensamiento crítico reflexivo en los estudiantes?
- **Interactividad:** ¿Cuánto se integran los estudiantes en el diálogo educativo en línea?
- **Soporte de profesores:** ¿Cómo capacitan los profesores a sus alumnos para participar en la educación en línea?
- **Soporte de los colegas:** El soporte proveído por los otros estudiantes, ¿es sensible y estimulante?
- **Interpretación:** Los estudiantes y los profesores, ¿tienen una apreciación correcta del otro a través de la comunicación en línea?

Independientemente de la visión dinámica del aprendizaje, es una nueva teoría del conocimiento: el constructivismo social, que asume al estudiante como un conceptualizador activo dentro del ambiente de aprendizaje social interactivo. Constructivismo social es una epistemología, o una forma de conocer, en la cual los estudiantes colaboran reflexivamente para co-construir nuevo entendimiento, especialmente dentro del contexto de la interrogación mutua basada en su propia experiencia.

Para que se de esta colaboración es vital el desarrollo de la capacidad de comunicación, es decir, la habilidad de integrarse en diálogo abierto y crítico con sus compañeros y profesores. Este diálogo se caracteriza por una enfática orientación a construir entendimiento recíproco, y por una actitud crítica frente a los supuestos que subyacen bajo los exámenes.

El COLLES se ha diseñado para monitorear la capacidad de explotar la capacidad interactiva del Internet y para integrar estudiantes en un ambiente de prácticas educativas dinámicas (esta información se ha adaptado de la página del COLLES. Si lo desea, puede encontrar más información en: <http://surveylearning.com/colles/>).

ATTLS - Attitudes to Thinking and Learning Survey (Encuesta sobre Actitudes hacia el Pensamiento y el Aprendizaje).

La teoría de las "formas de conocer", originaria del campo de la investigación de género (Belenky et al., 1986) nos ofrece una herramienta para examinar la calidad del discurso dentro de un ambiente de colaboración.

La Encuesta de Actitud sobre el Pensar y Aprender (ATTLS) es un instrumento desarrollado por Galiotti et al. (1999) para medir cuánto una persona es un "conocedor conectado" (CK, por sus siglas en inglés) o "conocedor desconectado" (SK).

Las personas con un CK más alto tienden a disfrutar más el aprendizaje y a menudo cooperan más, son congeniales, y tienen más libertad para construir sobre ideas ajenas, mientras que aquellos con SK más altos tienden a tomar una actitud de aprendizaje más crítica y argumentativa.

ACTIVIDADES 7.9



- Añada dos encuestas a su curso, una de ellas será de tipo COLLES y la otra de tipo ATTLS. Haga capturas de todo el proceso.

7.6.8 CHAT

El módulo de *chat* permite que los participantes discutan en tiempo real (sincrónico) a través de Internet. Esta es una manera útil de tener una comprensión de los otros y del tema en debate -usar una sala de *chat* es bastante diferente a utilizar los foros (diacrónicos)-. El módulo de *chat* contiene varias utilidades para administrar y revisar las conversaciones anteriores.

ACTIVIDADES 7.10



- Añada una sala de *chat* a su curso. Hacer capturas de todo el proceso.

7.6.9 GLOSARIO

Un glosario es una lista de definiciones, como un diccionario. Las entradas pueden buscarse o navegarse de diferentes maneras. El profesor puede utilizar glosarios cerrados, preparados por él, o glosarios abiertos en los que los estudiantes pueden contribuir sus definiciones. Estas definiciones pueden también ser calificadas por el profesor.

ACTIVIDADES 7.11



- Cree un nuevo glosario y añada al menos tres entradas a dicho glosario de nuestro curso Moodle. Haga capturas de todo el proceso.

7.6.10 LECCIÓN

Una lección proporciona contenidos de forma interesante y flexible. Consiste en una serie de páginas. Cada una de ellas normalmente termina con una pregunta y un número de respuestas posibles. Dependiendo de cuál sea la elección del estudiante, progresará a la próxima página o volverá a una página anterior. La navegación a través de la lección puede ser simple o compleja, dependiendo en gran medida de la estructura del material que se está presentando.

ACTIVIDADES 7.12



- Añada una nueva lección a su curso Moodle, vea las posibilidades que nos ofrece la lección y experimente con alguna de ellas. Haga capturas de todo el proceso seguido.

7.6.11 TALLER

El taller es una actividad para el trabajo en grupo con un vasto número de opciones. Básicamente, en un taller el profesor plantea una pregunta y los estudiantes preparan un pequeño trabajo en el que la responden. Entonces estos trabajos son distribuidos entre los participantes en el taller, para su evaluación crítica. Este módulo permite a los participantes diversas formas de evaluar los proyectos de los demás. El profesor puede dar también proyectos-prototipo. Se pueden coordinar la recopilación y distribución de esas evaluaciones de varias formas. Finalmente, el profesor puede calificar la actuación de cada estudiante, tanto en cuanto a su propio proyecto como en cuanto a cómo ha evaluado los proyectos de los demás. Recuerde que no todos los cursos utilizarán todos estos tipos de actividades. Eso depende del autor del curso, quien también podrá decidir cómo configurar exactamente cada una de las que utilice.

ACTIVIDADES 7.13



- Añada una actividad de tipo taller a su curso Moodle, vea las posibilidades que nos ofrece el taller y haga capturas de todo el proceso.

7.7 CREACIÓN DE CONTENIDOS

Algunos consejos importantes para los autores de cursos:

- ✓ Mientras un curso está aún en desarrollo, debe permanecer “invisible”. Esto se controla en la sección **Configuración del curso**. Además, debe crearse dentro de la sección **Cursos en pruebas**. Una vez esté listo para su uso por los alumnos se le mudará a la sección que le corresponda.
- ✓ Se debe elegir un nombre de curso que lo identifique claramente. Si hay más de un profesor en el mismo curso y se van a impartir programas diferentes, lo mejor es que cada uno genere un curso distinto. El nombre del curso debe permitir a los alumnos saber exactamente a qué se están inscribiendo.
- ✓ Se debe utilizar un código de curso que sea identificativo. Por ejemplo, usando las iniciales del nombre largo.
- ✓ En la descripción del curso que aparece junto con el nombre, es recomendable ser breves y concisos.

7.8 PERSONALIZACIÓN DE LA INTERFAZ

Los temas son paquetes que contienen archivos de configuración e imágenes. Moodle lee el paquete elegido y esto se traduce en un cambio de aspecto de la plataforma. Por suerte, cuando instalamos Moodle tenemos varios temas disponibles. Claro que si ninguno nos convence, podemos descargar más temas desde Internet o crear nuestro propio tema (esto último puede llegar a ser bastante complejo, dependiendo de lo que queramos).

Para empezar, desde el bloque **Administración** del sitio vamos a **Apariencia** y luego escogemos la opción **Temas**. Veremos dos ítems:

- ✓ *Ajustes de temas*. Permite establecer los parámetros de los temas, los que aparecen en la lista, si cada usuario puede elegir su propio tema, etc.
- ✓ *Selector de temas*. Es la lista de todos los temas disponibles en ese momento. Esa lista puede aumentar o disminuir si nosotros añadimos o quitamos temas de la carpeta correspondiente.

7.8.1 AJUSTES DE TEMAS

Ajustes de temas en Moodle (**Administración del sitio**, **Apariencias**, **Temas**, **Ajustes de temas**).

- **Lista de temas**. Podemos escribir los temas que queremos que se muestren en el selector y así ocultar el resto.
- **Permitir estilos al usuario**. Activar esta opción le permitirá a cada usuario tener un menú para cambiar la apariencia de Moodle (esa nueva apariencia será visible solo para él).
- **Permitir estilos en el curso**. Permite que cada curso tenga un tema (diseño) independiente. Aunque representa un gran trabajo crear un diseño por curso, la apariencia final es excelente. Por ejemplo, dentro del mismo sitio, si un alumno entra al curso de Historia verá un diseño alusivo al área, de forma similar en cursos como Biología, Literatura, entre otros.

- **Permitir temas por categorías.** Similar al caso anterior, pero ahora con categorías. Las categorías son grupos que contienen cursos. Ejemplos de categorías serían **niveles**, como categoría *cursos de primaria*, o categoría *cursos de secundaria*. O **áreas** como categoría *área de ciencias*, categoría *área de comunicación*, etc. La documentación de Moodle advierte que activar esta opción puede afectar el rendimiento del sistema.
- **Permitir a los usuarios ocultar bloques.** Los bloques laterales pueden ocultarse presionando un botón. Esta opción permite decidir si los usuarios tendrán o no esa opción.
- **Mostrar bloques en páginas de módulos.** Los módulos son funciones adicionales que permiten añadir actividades a Moodle. Esta opción permite decidir si a dichos módulos se les puede añadir o no bloques laterales.
- **Ocultar navegación de tipo de actividad.** Permite decidir si se verán los enlaces a las actividades a los usuarios.

Ajustes de temas

Lista de temas Valor por defecto: Vacío
themeist
Deje esta opción en blanco para permitir que se utilice cualquier tema válido. Si quiere acortar el menú de temas, puede especificar una lista de nombres separados por comas. Por ejemplo: standard,orangewhite

Permitir estilos al usuario ☐ Valor por defecto: No
allowuserthemes
Si se activa esta opción, los usuarios podrán elegir sus propios temas. Los temas de los usuarios anulan los temas del sitio (pero no los temas del curso)

Permitir estilos en el curso ☐ Valor por defecto: No
allowcoursethemes
Si activa esta opción, se permitirá a los cursos ajustar sus propios temas. Los temas de los cursos pasan por alto cualesquiera otras opciones de tema (sitio, usuario o sesión)

Permitir temas por categoría ☐ Valor por defecto: No
allowcategorythemes
Si activa esta opción, los temas pueden ajustarse al nivel de la categoría. Esto afectará a todas las categorías y cursos de nivel inferior ('hijos') a menos que tengan especificado su propio tema. ATENCIÓN: Habilitar esta opción puede afectar al rendimiento.

Permitir a los usuarios ocultar bloques ☒ Valor por defecto: Si
allowuserblockhiding
¿Desea que los usuarios puedan mostrar u ocultar bloques laterales en el sitio? Esta opción usa Javascript y cookies para recordar el estado de cada bloque colapsable, y sólo afecta al modo en que cada usuario ve la información.

Mostrar bloques en páginas de módulos ☐ Valor por defecto: No
showblocksmodpages
Algunos módulos de actividad admiten bloques en sus páginas. Si selecciona esta opción, los profesores podrán añadir bloques laterales a sus páginas. En caso contrario, la interfaz no mostrará esta característica.

Ocultar navegación de tipo de actividad Valor por defecto: No ocultar a nadie
hideactivitytypenavlink
Seleccione a quién ocultar el enlace al tipo de actividad (e.g., cuestionarios) en la navegación de los módulos de actividad.

Figura 7.36. Temas



Figura 7.37. Temas

7.8.2 SELECTOR DE TEMAS

Aquí podemos elegir un tema para nuestro sitio Moodle.

Se escoge el tema que nos agrade, le damos clic a **Elegir** y listo, ya tenemos un nuevo diseño para Moodle

ACTIVIDADES 7.14



➔ Personalice la interfaz de su curso Moodle. Realice capturas de todo el proceso seguido.

7.9 USUARIOS Y GRUPOS

La ubicación en Moodle de los roles de usuarios está en: **Administración del Sitio, Usuarios, Permisos, Definir roles.**

Los privilegios de usuario permiten restringir ciertas características o funcionalidades a un usuario en función de su nivel. Existen las siguientes categorías de usuarios, ordenadas de mayor a menor, en función del nivel de privilegios: administrador principal, administrador, creador de curso, profesor editor, profesor no editor, alumno e invitado.

7.9.1 ADMINISTRADOR

Un administrador en Moodle gestiona todo el sitio. Normalmente, el administrador supervisa la apariencia y la sensación que produce el Moodle de su organización y que lo hacen único.

El rol o categoría de administrador ocupa el nivel más alto en Moodle, en lo que a privilegios de usuario se refiere.

7.9.2 CREADOR DE CURSOS

La creación de nuevos cursos solo puede ser realizada por un usuario que tenga asignado ese privilegio. Un usuario que tenga asignado un rol de creador de cursos tiene un nivel de privilegios Moodle que le permite, además, asignar profesores y actuar como profesor con privilegios de edición. Un administrador tiene todos esos privilegios y más.

El creador del curso puede ser el profesor principal, el jefe de departamento o el coordinador del programa.

7.9.3 PROFESOR

Tiene control sobre un curso específico dentro de Moodle y la actividad de los Alumnos que están inscritos a él.

7.9.4 ESTUDIANTE

El rol de Estudiante es el tipo de usuario más básico de Moodle. Los estudiantes generalmente se matriculan en uno o varios cursos y acceden a Moodle. Los administradores y profesores deciden cómo se matriculan los estudiantes, así como lo que estos pueden ver y hacer dentro de Moodle. Esto puede variar en cada curso o en cada una de las muchas actividades de Moodle.

7.9.5 INVITADO

La gente puede acceder como invitado utilizando el botón **Entrar como invitado** en la pantalla de acceso (este botón puede ser habilitado o deshabilitado por el administrador). Para un curso dado, el profesor puede escoger si permite o no el acceso a invitados. Los invitados tienen SIEMPRE acceso de “solo-lectura”, lo que significa que no pueden enviar mensajes o entorpecer el curso a los verdaderos alumnos. Los invitados no pueden:

- ✓ Enviar mensajes a discusiones.
- ✓ Editar páginas wiki.
- ✓ Contestar cuestionarios.
- ✓ Enviar tareas.
- ✓ Contribuir al contenido o comentarios de los glosarios.
- ✓ Ver el contenido de los paquetes SCORM (porque se realiza un seguimiento).

ACTIVIDADES 7.15



- Añada al menos cinco usuarios a nuestro curso. Uno de ellos debe tener perfil de profesor, al resto les asignaremos perfil de estudiante. Realice capturas de todo el proceso.

7.10 CONTROL DE ACCESOS. RED MOODLE

La Red Moodle es una característica nueva que se puede encontrar a partir de la versión 1.8 de Moodle. Esta característica permite a un administrador de Moodle establecer un enlace con otro Moodle, y compartir algunos recursos de ese otro Moodle.

La publicación inicial de la Red Moodle se acompaña de un nuevo *plugin* de autenticación que hace que sea posible el inicio de sesión único (*single sign on*) entre sitios Moodle. Un usuario con el nombre de usuario *jody* inicia la sesión en su servidor Moodle como de costumbre y pulsa en un enlace que le lleva a una página en otro servidor Moodle. En el caso habitual, *jody* debería tener solo los privilegios del usuario invitado en el servidor Moodle remoto, pero entre bastidores la validación única ha establecido una sesión de autenticación completa en el sitio remoto.

La característica de la Red Moodle necesita que su servidor tenga las extensiones Curl y OpenSSL de PHP instaladas. Cuando instale o actualice a Moodle 1.8, su sistema creará un nuevo certificado OpenSSL para cifrar las comunicaciones con otros Moodle, y rotará las claves de cifrado una vez al mes (aproximadamente) a partir de ese momento.

La comunicación se lleva a cabo por medio de un transporte XML-RPC y los documentos XML-RPC se envuelven primero en un sobre XMLDSIG (*XML digital signature*) y, posteriormente, en un sobre XMLENC (*XML encryption*). El cifrado se realiza completamente dentro de PHP y no se necesita un servidor *https* (Apache SSL).

Se puede habilitar un modo especial que permite a una máquina con una dirección IP específica hacer llamadas a la capa XML-RPC sin usar sobres ni de cifrado ni de firma. Se proporciona este modo para permitir a Moodle comunicar con otros sistemas software en los que la integración de las firmas y el cifrado pudiera ser excepcionalmente difícil. Sin embargo, no se prevé que nunca se llegue a habilitar la comunicación no cifrada entre sitios Moodle.

7.11 INTEGRACIÓN DE MÓDULOS

7.11.1 MÓDULO DE TAREAS

- Puede especificarse la fecha final de entrega de una tarea y la calificación máxima que se le podrá asignar.
- Los estudiantes pueden subir sus tareas (en cualquier formato de archivo) al servidor. Se registra la fecha en que se han subido.
- Se permite enviar tareas fuera de tiempo, pero el profesor puede ver claramente el tiempo de retraso.
- Para cada tarea en particular, puede evaluarse a la clase entera (calificaciones y comentarios) en una única página con un único formulario.
- Las observaciones del profesor se adjuntan a la página de la tarea de cada estudiante y se le envía un mensaje de notificación.
- El profesor tiene la posibilidad de permitir el reenvío de una tarea tras su calificación.

7.11.2 MÓDULO DE CONSULTA

Es como una votación. Puede usarse para votar sobre algo o para recibir una respuesta de cada estudiante (por ejemplo, para pedir su consentimiento para algo).

- El profesor puede ver una tabla que presenta de forma intuitiva la información sobre quién ha elegido qué.
- Se puede permitir que los estudiantes vean un gráfico actualizado de los resultados.

7.11.3 MÓDULO FORO

Hay diferentes tipos de foros disponibles: exclusivos para los profesores, de noticias del curso y abiertos a todos.

- Todos los mensajes llevan adjunta la foto del autor.
- Las discusiones pueden verse anidadas, por rama, o presentar los mensajes más antiguos o los más nuevos primeros.
- El profesor puede obligar la suscripción de todos a un foro o permitir que cada persona elija a qué foros suscribirse de manera que se le envíe una copia de los mensajes por correo electrónico.
- El profesor puede elegir que no se permitan respuestas en un foro (por ejemplo, para crear un foro dedicado a anuncios).
- El profesor puede mover fácilmente los temas de discusión entre distintos foros.

7.11.4 MÓDULO DIARIO

Los diarios constituyen información privada entre el estudiante y el profesor.

- Cada entrada en el diario puede estar motivada por una pregunta abierta.
- La clase entera puede ser evaluada en una página con un único formulario, por cada entrada particular de diario.
- Los comentarios del profesor se adjuntan a la página de entrada del diario y se envía por correo la notificación.

7.11.5 MÓDULO CUESTIONARIO

- Los profesores pueden definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios.
 - Las preguntas pueden ser almacenadas en categorías de fácil acceso y estas categorías pueden ser “publicadas” para hacerlas accesibles desde cualquier curso del sitio.
 - Los cuestionarios se califican automáticamente y pueden ser recalificados si se modifican las preguntas.
- Los cuestionarios pueden tener un límite de tiempo a partir del cual no estarán disponibles.
- El profesor puede determinar si los cuestionarios pueden ser resueltos varias veces y si se mostrarán o no las respuestas correctas y los comentarios
- Las preguntas y las respuestas de los cuestionarios pueden ser mezcladas (aleatoriamente) para disminuir las copias entre los alumnos.
- Las preguntas pueden crearse en HTML y con imágenes.
- Las preguntas pueden importarse desde archivos de texto externos.
- Las preguntas pueden tener diferentes métricas y tipos de captura.

7.11.6 MÓDULO RECURSO

- Admite la presentación de un importante número de contenido digital, Word, PowerPoint, Flash, vídeo, sonidos, etc.
- Los archivos pueden subirse y manejarse en el servidor o pueden ser creados sobre la marcha usando formularios web (de texto o HTML).
- Pueden enlazarse aplicaciones web para transferir datos.

7.11.7 MÓDULO ENCUESTA

- Se proporcionan encuestas ya preparadas (COLLES, ATTLS) y contrastadas como instrumentos para el análisis de las clases en línea.
- Se pueden generar informes de las encuestas, los cuales incluyen gráficos. Los datos pueden descargarse con formato de hoja de cálculo Excel o como archivo de texto CSV.
- La interfaz de las encuestas impide la posibilidad de que sean respondidas solo parcialmente.
- A cada estudiante se le informa sobre sus resultados comparados con la media de la clase.

7.11.8 MÓDULO WIKI

- El profesor puede crear este módulo para que los alumnos trabajen en grupo en un mismo documento.
- Todos los alumnos podrán modificar el contenido incluido por el resto de compañeros.
- De este modo cada alumno puede modificar la wiki del grupo al que pertenece, pero podrá consultar todas las wikis.

7.12 COPIAS DE SEGURIDAD

Las copias de seguridad están dentro de **Administración del curso**, **Copia de seguridad** (Figura 7.38). Debemos marcar aquello sobre lo que queremos hacer la copia de seguridad (Figura 7.39). Una vez seleccionado pulsamos sobre **Siguiente**.

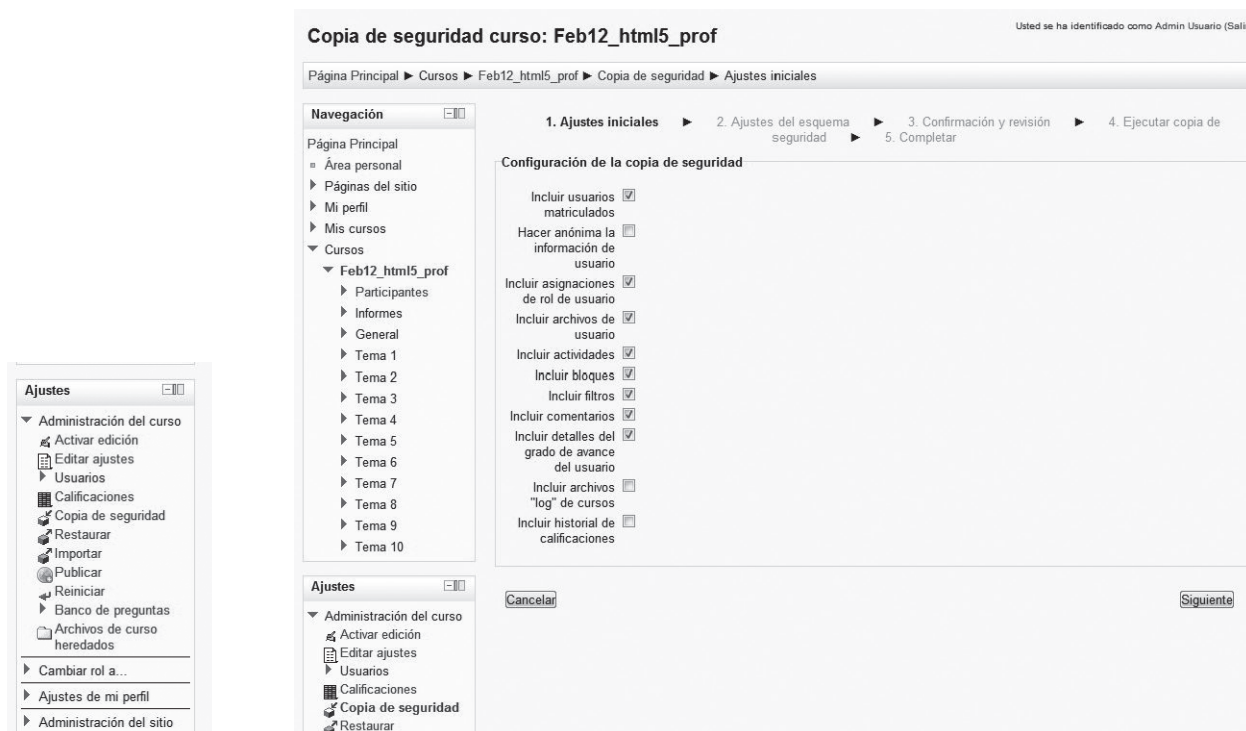


Figura 7.38. Copias

Figura 7.39. Copias

En la siguiente ventana podemos personalizar sobre qué material vamos a realizar la copia de seguridad (Figura 7.40). En la siguiente pantalla pondremos nombre al archivo de la copia de seguridad (Figura 7.41).

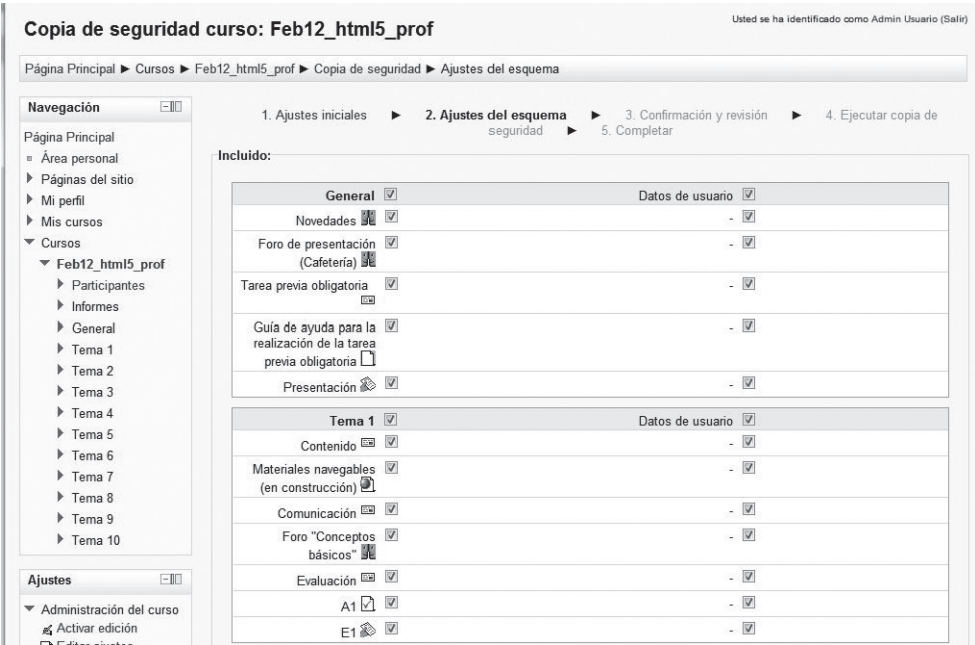


Figura 7.40. Copias

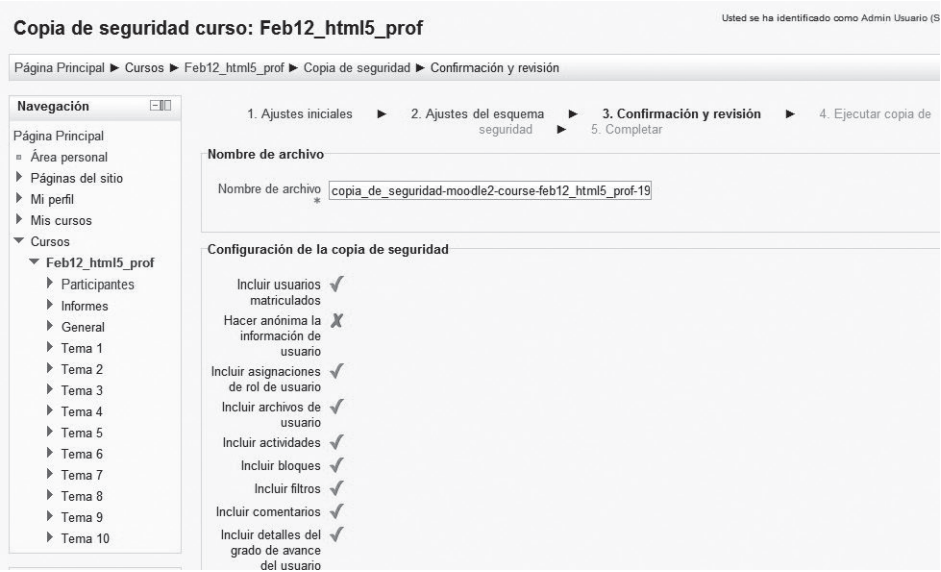


Figura 7.41. Copias

Ya podemos pulsar sobre **Ejecutar copia de seguridad** (Figura 7.42) y se realizará la copia seleccionada. Una vez finalizada, nos aparecerá un mensaje indicando su fin (Figura 7.43). La copia de seguridad se efectua en el propio servidor. Si queremos podemos descargarla a nuestro equipo (Figura 7.44).

Este curso, y sus materiales, se distribuyen c... ✓

Tema 11 ✓	Datos de usuario ✓
Foro de tutoría ✓	- ✓
Guía de la Tutoría ✓	- ✓
Guía de la Tutoría en PDF ✓	- ✓
Información sobre finalización del curso ✓	- ✓
Modificación de datos administrativos ✓	- ✓
El Aula Virtual del ITE ✓	- ✓
El Aula Virtual del ITE en PDF ✓	- ✓
Videotutorial "Comprimir archivos" ✓	- ✓
Videotutorial "Subir un archivo" ✓	- ✓
Sobre licencias ✓	- ✓
Sobre licencias. PDF ✓	- ✓

Tema 12 ✓	Datos de usuario ✓
-----------	--------------------

Tema 13 ✓	Datos de usuario ✓
-----------	--------------------

[Anterior](#) [Cancelar](#) [Ejecutar copia de seguridad](#)

En este formulario hay campos obligatorios*

Figura 7.42. Copias

tml5_prof ► Copia de seguridad ► Completar

1. Ajustes iniciales ► 2. Ajustes del esquema seguridad ► 3. Confirmación y revisión ► 4. Ejecutar copia de seguridad ► 5. Completar

El archivo de copia de seguridad se creó con éxito

[Continuar](#)

Figura 7.43. Copias

Zona de copia de seguridad de curso ③				
Nombre de archivo	Fecha	Tamaño	Descargar	Restaurar
copia_de_seguridad-moodle2-course-feb12_html5_prof-19012012-1042.mbz	jueves, 19 de enero de 2012, 10:47	167.5Mb	Descargar	Restaurar

[Gestionar archivos de copia de seguridad](#)

Zona de copia de seguridad privada de usuario ③				
Nombre de archivo	Fecha	Tamaño	Descargar	Restaurar

Figura 7.44. Copias

Mediante la copia de seguridad vamos a poder exportar los datos de un curso para llevárnoslos a otro sitio (por ejemplo, preparar los datos en un servidor local y una vez estén terminados los exportamos a un servidor web).

ACTIVIDADES 7.16



- Realice una copia de seguridad de su curso Moodle al finalizar descarga el fichero en su equipo. Haga capturas de todo el proceso seguido.



RESUMEN DEL CAPÍTULO

En este capítulo tratamos con una de las herramientas más utilizadas sobre todo por los docentes de informática y que poco a poco va tomando mucha repercusión sobre el resto.

Moodle es una potentísima herramienta de teleformación que nos permite crear cursos de manera muy sencilla y con muy pocos conocimientos de informática. Esto ha hecho que, como se comentaba en el anterior párrafo, gran parte del profesorado utilice esta herramienta para complementar y ampliar el uso de libros de texto gracias a la posibilidad de introducir elementos multimedia, espacio para que los alumnos puedan subir las actividades encomendadas, cuestionarios, etc.



EJERCICIOS PROPUESTOS

- 1. Defina con sus palabras qué es moodle y para qué sirve.
- 2. La inmensa mayoría de páginas web con teleformación o formación a distancia están implementadas en Moodle. ¿Esto a qué es debido?
- 3. Localice en Internet páginas con teleformación en Moodle. Ponga al menos tres ejemplos de páginas de este tipo.
- 4. De las actividades, ¿cuál de ellas le ha gustado más? ¿Por qué?
- 5. ¿Hay otros gestores de contenidos para *e-learning* que no son Moodle? Si la respuesta es afirmativa, indique algún ejemplo.



TEST DE CONOCIMIENTOS

1 Moodle aparece en su versión 1.0 en el año:

- a) 1980.
- b) 1987.
- c) 1995.
- d) 2006.
- e) 2011.
- f) Todas son ciertas.
- g) Todas son falsas.

2 Moodle es:

- a) Gratuito.
- b) Gratis los 3 primeros cursos, a partir del cuarto es de pago.
- c) De pago.
- d) De pago solo para empresas, para los usuarios comunes es gratis.
- e) Todas son falsas.

3 Para instalar Moodle necesito tener:

- a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
- b) Un servidor web con soporte PHP y MySQL.
- c) Un equipo con Windows.
- d) Un equipo con Linux.
- e) Todas son ciertas.
- f) Todas son falsas.

4 Para utilizar Moodle necesito tener:

- a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
- b) Un equipo normal con una conexión normal.
- c) Un equipo con Windows.
- d) Un equipo con Linux.
- e) Todas son ciertas.
- f) Todas son falsas.

5 Cualquier persona puede acceder a un curso de Moodle creado:

- a) Por supuesto, ya que tiene licencia GPL, cualquier persona en la Red podrá acceder a nuestro curso.
- b) Si no está matriculado en el curso, nunca podrá acceder al mismo.
- c) Si se permite el acceso a invitados podrá entrar pero no podrá participar en el mismo.
- d) Las respuestas a y b son ciertas.
- e) Las respuestas b y c son ciertas.
- f) Todas son falsas.

8

Foros: BbPress

OBJETIVOS DEL CAPÍTULO

- ✓ Definir el concepto de foro.
- ✓ Diferenciar los componentes de un foro.
- ✓ Conocer y utilizar uno de los foros con licencia GPL que más fuerza están tomando.

8.1 DEFINICIÓN

Los foros en Internet son también conocidos como foros de mensajes, de opinión o foros de discusión y son una aplicación web que le da soporte a discusiones u opiniones en línea. Son los descendientes modernos de los sistemas de noticias BBS (*Bulletin Board System*) y Usenet, muy populares en los años 1980 y 1990. Por lo general, los foros en Internet existen como un complemento a un sitio web invitando a los usuarios a discutir o compartir información relevante a la temática del sitio, en discusión libre e informal, con lo cual se llega a formar una comunidad en torno a un interés común. Las discusiones suelen ser moderadas por un coordinador o dinamizador quien generalmente introduce el tema, formula la primera pregunta, estimula y guía, sin presionar, otorga la palabra, pide fundamentaciones y explica lo expuesto antes de cerrar la discusión.

Comparado con las wikis, no se pueden modificar los aportes de otros miembros a menos que tenga ciertos permisos especiales como moderador o administrador. Por otro lado, comparado con los *weblogs*, se diferencian porque los foros permiten una gran cantidad de usuarios y las discusiones están añadidas, algo similar a lo que serán los comentarios en los *weblogs*. Además, por lo general, los foros suelen ser de temas más diversos o amplios con una cantidad de contenido más variado y la posibilidad de personalizar a nivel usuario.

Un foro en Internet permite que el administrador del sitio defina varios foros sobre una sola plataforma. Estos funcionaron como contenedores de las discusiones que empezarán los usuarios; otros usuarios pueden responder en las discusiones ya comenzadas o empezar unas nuevas según lo crean conveniente.

En este capítulo nos vamos a centrar sobre bbPress, una herramienta de los creadores de WordPress, nueva y gratuita para la creación de foros. BbPress se centra en los estándares web, la facilidad de uso, facilidad de integración, y la velocidad. La primera versión de bbPress aparece en el año 2008.

La intención de este CMS es que sea tan pequeño y ligero como sea posible al tiempo que permita añadir nuevas características a través de un gran sistema de *plugins*. ¿Qué significa todo esto? BbPress está dispuesto a asumir cualquier tipo de trabajo. Así que vamos a empezar.

8.2 LICENCIAS DE USO DE BBPRESS

bbPress es un sistema de gestión de foros con licencia GPL que comparte código con WordPress (mediante Backpress), de modo que se pueden integrar ambos sistemas de forma muy fácil. Requiere ser instalado en un servidor web propio.

8.3 REQUERIMIENTOS DE FUNCIONAMIENTO

- Un servidor web que soporte PHP y MySQL. Se recomienda utilizar Apache o Litespeed como servidores web.
- PHP versión 4.3 o superior.
- MySQL versión 4.0 o superior.

8.4 INSTALACIÓN

Los pasos para la instalación de bbPress son los siguientes:

- 1 Descargamos la aplicación bbPress desde su página web: <http://bbpress.org/download/> (Figura 8.1).

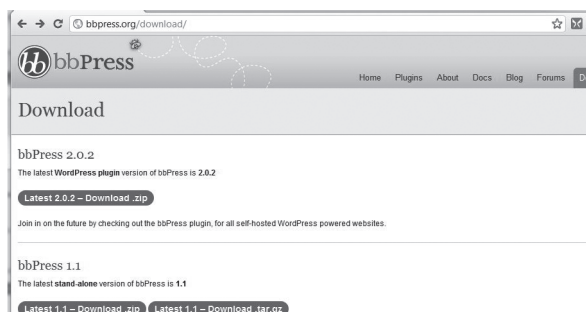


Figura 8.1. Descarga de bbPress

- 2 Subimos los ficheros descomprimidos a nuestro servidor, también podemos utilizar un servidor local (tipo *xampp*) para hacer las pruebas oportunas. También podemos subir los ficheros de idioma al servidor (opcional, aunque recomendable) (Figura 8.2).

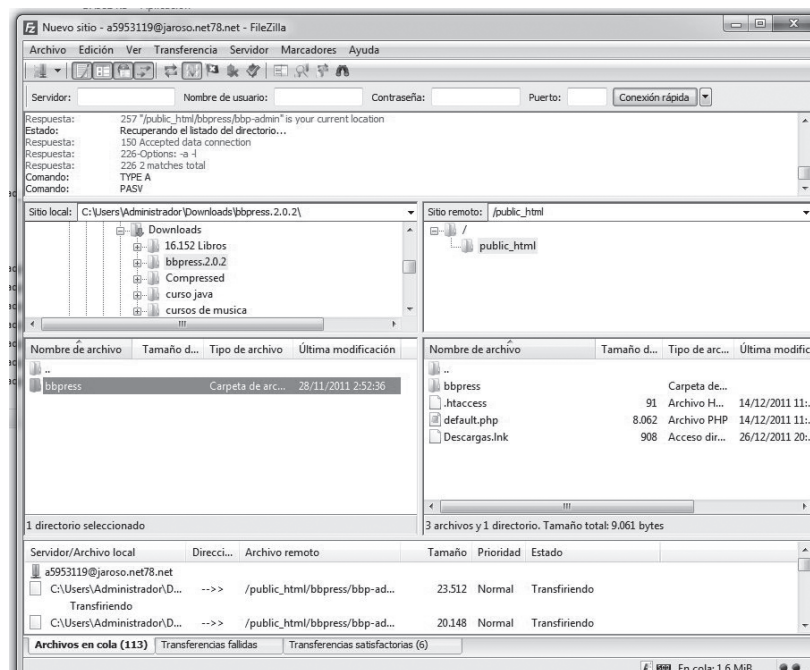


Figura 8.2. Subir al servidor

3 Antes de realizar el proceso de instalación debemos crear una base de datos para utilizarlo con la aplicación (Figura 8.3).

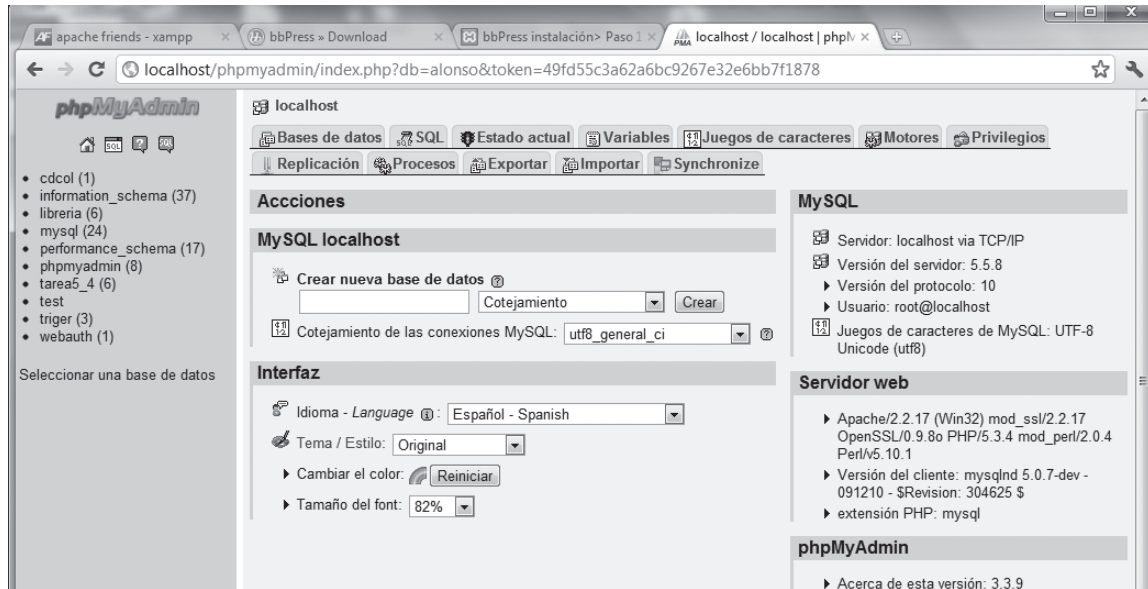


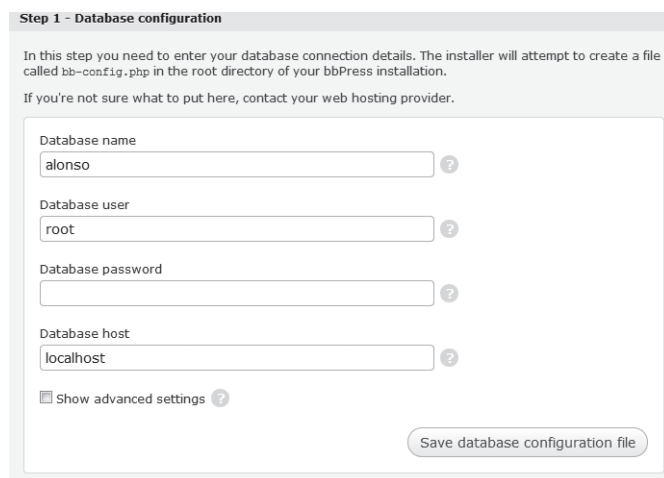
Figura 8.3. Creamos la base de datos

4 Accedemos a la página inicial del sitio e instalamos bbPress siguiendo las instrucciones del instalador (Figura 8.4).



Figura 8.4. Lanzamos la instalación

5 En el apartado de configuración de la base de datos debemos rellenar varios campos: el nombre de la base de datos escribiremos en nombre de la base que hayamos creado. A su vez, deberemos rellenar el nombre de usuario y la contraseña que son los del gestor de la base de datos (Figura 8.5).



Step 1 - Database configuration

In this step you need to enter your database connection details. The installer will attempt to create a file called `bb-config.php` in the root directory of your bbPress installation.

If you're not sure what to put here, contact your web hosting provider.

Database name
alonso ?

Database user
root ?

Database password
? ?

Database host
localhost ?

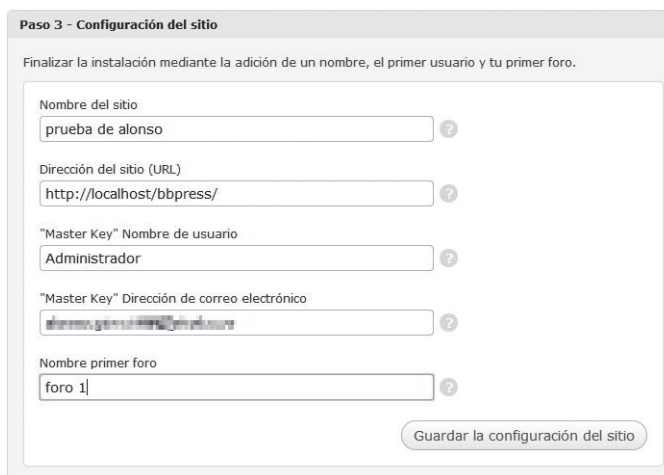
☐ Show advanced settings ?

Save database configuration file

Figura 8.5. Configuración de la base de datos

6 El siguiente paso es para la integración con WordPress, este paso es opcional y no lo vamos a utilizar.

7 Ahora configuraremos el foro propiamente dicho, dotaremos de nombre a dicho foro, asignaremos la dirección web de dicho foro y crearemos un usuario administrador (llamado *master key*), que será el amo todopoderoso del foro. A su vez también crearemos el primer foro de discusión de nuestro nuevo sitio (Figura 8.6).



Paso 3 - Configuración del sitio

Finalizar la instalación mediante la adición de un nombre, el primer usuario y tu primer foro.

Nombre del sitio
prueba de alonso ?

Dirección del sitio (URL)
http://localhost/bbpress/ ?

"Master Key" Nombre de usuario
Administrador ?

"Master Key" Dirección de correo electrónico
? ?

Nombre primer foro
foro 1 ?

Guardar la configuración del sitio

Figura 8.6. Configuración del sitio

8 Después de hacer este trámite nos aparecerá un mensaje informándonos de que se guardarán los datos que se han creado y pulsaremos en el botón **Completar la instalación**. Y, muy importante, al terminar nos saldrá una pantalla con la información final de su instalación; en este momento le generará la contraseña de su usuario (Figura 8.7).

Instalación se ha completado con algunos errores!

Su instalación se ha completado con algunos errores menores. Vea el registro de errores a continuación para obtener información más específica.

Ahora puede iniciar sesión con los siguientes datos:

Nombre de Usuario:
alonso

Contraseña:
whKh0bHb TIAS%

Dirección del sitio:
http://localhost/bbpress/

Figura 8.7. Instalación completada

ACTIVIDADES 8.1



» Descargue e instale la última versión de bbPress.

8.5 ESTRUCTURA

Partimos de la página principal del foro:

Register or log in - lost password?

Nombre de usuario

Contraseña

Log in »

☐ Remember me

Search »

foro jaroso
Just another bbPress community

Hot Tags

bbPress

Latest Discussions

Tema — Add New »	Mensajes	Último publicador	Novedad
Tu primer tema	1	juan	1 day

Foros

Main Theme	Temas	Mensajes
dudas	1	1

foro jaroso is proudly powered by bbPress.

Figura 8.8. Estructura de bbPress

- **Zona de registro-autenticación de usuarios:** en esta zona podemos registrarnos en el foro o identificarnos en el caso de que ya estemos registrados en el foro (Figura 8.9).

Figura 8.9. Zona de registro-autenticación

- **Nombre del foro:** el nombre del foro que pusimos en la instalación (Figura 8.10).



Figura 8.10. Nombre del foro

- **Búsqueda:** permite localizar temas y mensajes en el foro (Figura 8.11).

Figura 8.11. Búsqueda

- **Últimos post publicados:** últimas entradas al foro (Figura 8.12).

Latest Discussions			
Tema — Add New »	Mensajes	Último publicador	Novedad
Tu primer tema	1	juan	1 day

Figura 8.12. Ultimos post

- **Foros disponibles:** los temas de discusión que existen y están disponibles actualmente en nuestro foro (Figura 8.13).

Foros		
Main Theme	Temas	Mensajes
dudas	1	1

Figura 8.13. Foros disponibles

ACTIVIDADES 8.2



- Haga una captura de los diferentes elementos que componen su foro recién creado: zona de registro, nombre del foro, búsqueda, último post y foros disponibles.

8.6 PERSONALIZACIÓN DE LA INTERFAZ

Para cambiar la interfaz, debemos estar identificados como *master key* (Figura 8.14). Una vez identificados debemos pulsar sobre **Admin** (Figura 8.15).

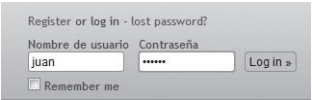


Figura 8.14. Identificación

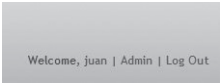


Figura 8.15. Admin

Seleccionaremos **Appearance** (Figura 8.16). Nos aparecerá un menú con los temas disponibles, deberemos elegir el que más nos guste y activarlo con la opción **Activate** (Figura 8.17).



Figura 8.16. Appearance

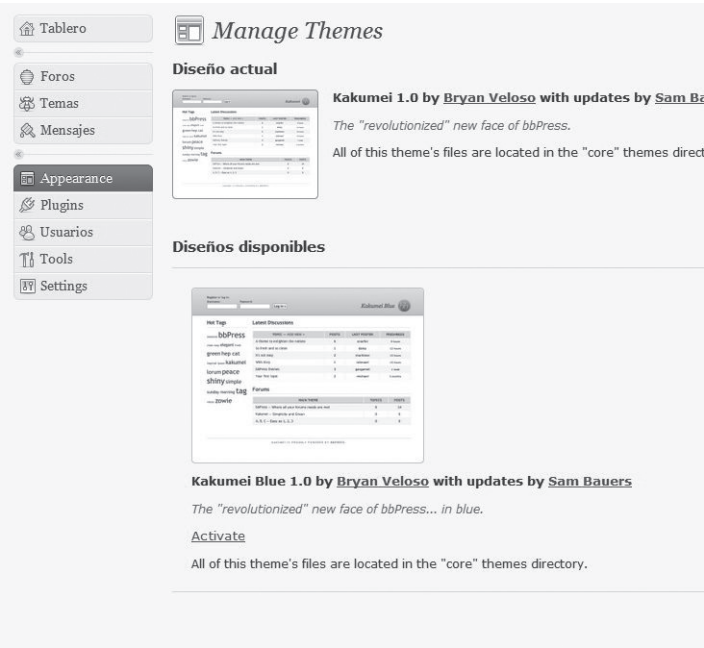


Figura 8.17. Manage Themes

ACTIVIDADES 8.3



➤ Personalice la interfaz de su foro.

8.7 PERFILES DE USUARIOS Y MECANISMOS DE SEGURIDAD INTEGRADOS

En este apartado se puede apreciar la seguridad referente a los diferentes tipos de usuario que se mencionaron anteriormente en el apartado Seguridad. Los diferentes tipos de usuario y su seguridad serían los siguientes:

8.7.1 MASTER KEY

Es el usuario con más privilegios. Este tipo de usuarios puede realizar todo tipo de operaciones. Eliminar foros, eliminar usuarios... (incluidos los administradores).

8.7.2 ADMINISTRADOR

Es el usuario con más privilegios después de *Master Key*. Este tipo de usuario es el encargado gestionar el foro pero tiene algunas restricciones que lo diferencian con el *master key* ya que estos no pueden modificar lo creado por otro administrador, es decir, eliminar, modificar, bloquear todo que haya creado otro administrador.

8.7.3 MODERADOR

Este usuario tiene los privilegios de poder controlar todo lo relacionado con su foro, es decir, solamente en el foro que este asignado. No podrá controlar la conversación de otros foros como podrían los usuarios anteriores.

8.7.4 MIEMBRO

Es aquel usuario que pertenece al foro pero no tiene ningún tipo de privilegios con el resto de los usuarios.

8.7.5 BLOQUEADO

Este usuario es un miembro que ha sido inutilizado por un usuario con mayor privilegio debido a que haya incumplido alguna de las normas a seguir del foro.

8.8 USUARIOS Y GRUPOS

En primer lugar se recomienda cambiar la contraseña del usuario creado como *master key* para poner una más fácil de recordar.

Con el usuario identificado nos dirigiremos al enlace **Admin** (Figura 8.18). Dentro de éste, al apartado **Usuarios** (Figura 8.19).

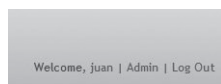


Figura 8.18. Admin

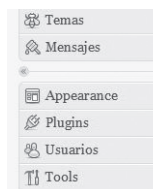


Figura 8.19. Usuarios

Seleccionamos nuestro usuario y pulsamos sobre **Editar** (Figura 8.20). En el apartado **New Password** (nueva contraseña) cambiamos la contraseña y guardamos los cambios (Figura 8.21).



Figura 8.20. Editar

Figura 8.21. New Password

Para añadir un nuevo usuario, dentro de **Usuarios**, haremos lo siguiente:

En el menú de usuarios, pulsamos sobre **Add New** (añadir nuevo) (Figura 8.22). En este apartado rellenaremos los datos personales. Lo más importante son el nombre de usuario, con el cual el usuario se identificará en el sistema, la contraseña y el tipo de usuario que será. Existen muchos tipos de usuarios: bloqueados, inactivos, moderadores, administradores. En nuestro caso hemos elegido el perfil **Miembro** (Figura 8.23).

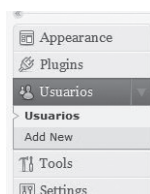
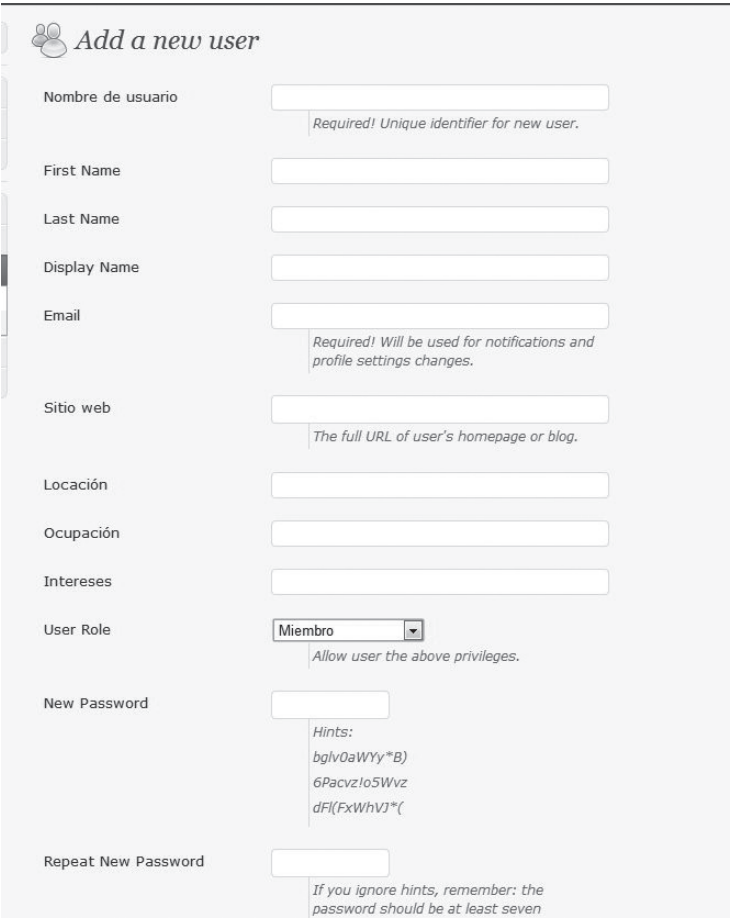


Figura 8.22. Usuarios, Add New



 **Add a new user**

Nombre de usuario
Required! Unique identifier for new user.

First Name

Last Name

Display Name

Email
Required! Will be used for notifications and profile settings changes.

Sitio web
The full URL of user's homepage or blog.

Locación

Ocupación

Intereses

User Role
Allow user the above privileges.

New Password
Hints:
bglv0aWYy*B)
6Pacvz!o5Wvz
dFl(FxWhVJ*{(

Repeat New Password
If you ignore hints, remember: the password should be at least seven

Figura 8.23. Añadimos nuevo usuario

ACTIVIDADES 8.4



- Añadir tres usuarios a nuestro foro: uno de ellos con perfil administrador, otro con perfil de usuario y el último con perfil de moderador.

8.9 INTEGRACIÓN DE MÓDULOS

La herramienta da la opción de poder integrarse en un WordPress. Este proceso era el paso número dos de la instalación que recordamos que era opcional, para la integración de esta utilidad en WordPress deberíamos de seguir el siguiente procedimiento:

Para bbPress se debe trabajar con la versión más reciente de WordPress 2.9.x. Debe actualizarse el *blog* antes de trabajar con WordPress y bbPress.

8.9.1 BASE DE DATOS

BbPress pueden compartir datos de los usuarios con una instalación de WordPress. Todo lo que necesita hacer es especificar la ubicación de la base de datos sobre la instalación o después de la **Integración de WordPress** del área **Configuración** en su administración bbPress.

8.9.2 COOKIES

La integración de las *cookies* permite a los usuarios iniciar sesión en cualquiera de su *blog* de WordPress o de sus foros bbPress. Esta función requiere WordPress y bbPress instalados en el mismo nombre de dominio, o por lo menos en los subdominios del mismo nombre de dominio.

Se debe consultar la sección **Integración de WordPress** del área **Configuración** en su administración bbPress para los ajustes que necesita añadir a su fichero de configuración de WordPress, después de haber agregado la información de WordPress en el de **Cookies**.

8.9.3 REGISTRO

Cuando un usuario se registra en el foro bbPress, su **blog** de WordPress no sabe nada de él (y viceversa). El usuario será creado, pero WordPress no se sabe qué tipo de usuario se conecta.

En la **Integración de WordPress** del área **Configuración**, en la administración bbPress, se encuentra un **Mapa de función de usuario**, donde se pueden establecer los roles que se aplicará a los usuarios que se registren en sus foros. Los usuarios que se registren en el *blog* de WordPress tendrán el papel apropiado aplicado a su cuenta la primera vez que acceder a bbPress.

Si no se configuran los roles de usuarios no funcionará. Al hacer un mapa de configuración de rol (o cambiar) los usuarios que faltan a uno u otro rol se actualizará, pero los roles existentes no se verán afectados.

8.9.4 FUNCIONES (OPCIONAL)

BbPress no tendrá acceso a las funciones de WordPress de forma manual a menos que se configure bbPress para cargar WordPress en primer lugar. Este paso de integración también se conoce como **integración profunda**, que es una parte totalmente opcional de integración y no se requiere para obtener las *cookies* compartidas e inicios de sesión de trabajo. Con el fin de hacerlo, es necesario poner *require_once* (*dirname (FILE) ' / ruta / a / wp-load.php*.); en bbPress *'bb-config.php* (*wp-load.php* está en el mismo directorio como WordPress *'wp-config.php* archivo, así que si la carpeta de bbPress está directamente dentro de la carpeta WordPress, el comando se convierte en *require_once* (*dirname (FILE) ' / .. / wp-load.php*. ');).

Hacer esto le añade mucho peso a la instalación de bbPress ya que esto haría WordPress para cargar en su totalidad. Su mejor opción es tratar de emular la funcionalidad que requieren dentro de un *plugin* bbPress.

8.10 GESTIÓN DE TEMAS

Para crear un nuevo tema para debatir en el foro realizaremos la siguiente ruta: **Admin, Forums, Add forum**; y nos aparecerá la siguiente pantalla.

Figura 8.24. Temas

ACTIVIDADES 8.5



- Añada tres nuevos temas a nuestro foro.

8.11 PUBLICACIÓN

En cuanto a la publicación de un comentario en los foros creados, su elaboración es muy sencilla simplemente hay que ir a la página principal de bbPress seleccionar un tema de los creados y aparecerá el apartado para poder enviar un nuevo *post* o comentario.

Figura 8.25. *Publicación*

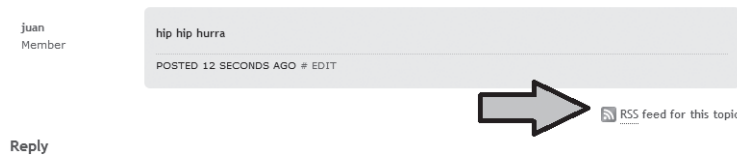


Figura 8.26. *Publicación*

ACTIVIDADES 8.6



- Añada dos nuevos comentarios a cada uno de los temas creados en la práctica anterior.

8.12 SINDICACIÓN DE CONTENIDOS

Una vez publicado el comentario podemos ver también un icono RSS para poder seguir todo lo referente al foro. RSS son las siglas de *Really Simple Syndication*, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador). A pesar de eso, es posible utilizar el mismo navegador para ver los contenidos RSS.

8.13 COPIAS DE SEGURIDAD

Para hacer una copia de seguridad de todo lo hecho hasta ahora introducimos en el navegador la dirección de nuestro servidor de bases de datos PHPMyAdmin (Figura 8.27).

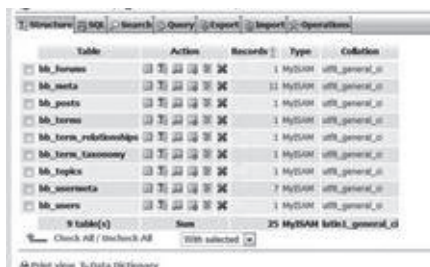


Figura 8.27. phpMyAdmin

Seleccionamos las tablas de la base de datos que hemos creado y acto seguido la opción **export** (Figura 8.28).

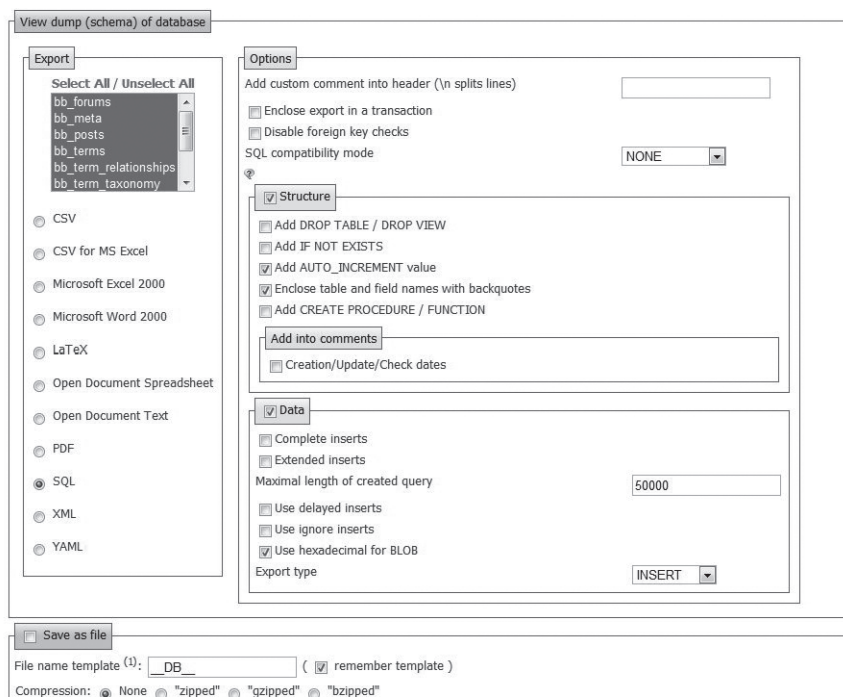


Figura 8.28. Exportar

Marcamos **Save as file** y pulsamos sobre **Go**. Debemos guardar el fichero `.sql` que crea, pues es la copia de seguridad de la base de datos.

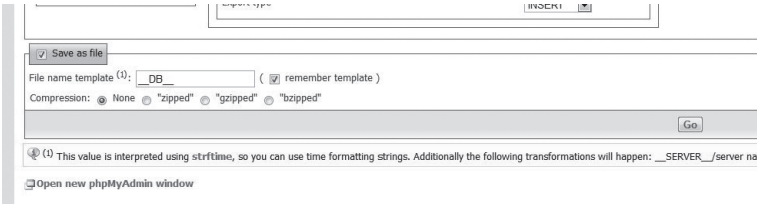


Figura 8.29. Hacer copia de seguridad

ACTIVIDADES 8.7



➤ Hacer una copia de seguridad del contenido de nuestra base de datos creada para nuestro foro.

8.14 SELECCIÓN DE MODIFICACIONES A REALIZAR

Para poder modificar un comentario escrito anteriormente hay que ir al comentario escrito y nos aparecerá la opción **EDIT** si queremos modificar el comentario que se subido anteriormente (Figura 8.30).

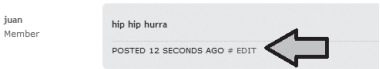


Figura 8.30. Editar

Si somos un usuario con privilegios también tendremos la opción de borrar el comentario (Figura 8.31):



Figura 8.31. Borrar

ACTIVIDADES 8.8



➤ De los comentarios insertados en la Actividad 8.6, elimine 2 de ellos y modifique otros 2. Haga capturas de todo el proceso seguido.

8.15 INCORPORACIÓN Y ADAPTACIÓN DE FUNCIONALIDADES

A la hora de poder incorporar o adaptar nuevas funcionalidades dentro de **Admin**, en la pestaña **Plugins** hay una variedad de **añadidos** para poder mejorar nuestro servidor de foros.

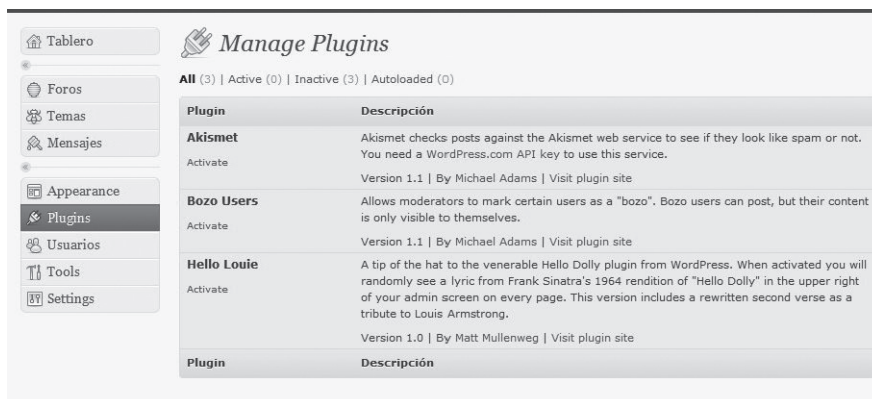


Figura 8.32. Plugins

ACTIVIDADES 8.9



➤ Instale un *plugin* que crea que es interesante para su foro. Haga capturas de todo el proceso seguido.



RESUMEN DEL CAPÍTULO

Los foros son una herramienta muy utilizada en la actualidad, hay foros de diversos temas: de hardware, de belleza, incluso hay algunos muy populares donde los usuarios añaden *links* de descarga para películas, series y programas.

En este capítulo nos hemos centrado en bbPress. BbPress es una herramienta “joven” pero con mucho potencial, debido principalmente a que es gratuita y extremadamente sencilla de utilizar. BbPress ha sido creado por los autores de WordPress y además de permitir la integración con *blogs* creados con dicha herramienta, permite también un funcionamiento en solitario.

A lo largo del capítulo hemos estudiado cómo descargarlo, instalarlo y configurarlo para que los usuarios puedan hacer sus aportaciones a nuestro foro.



EJERCICIOS PROPUESTOS

- 1. Defina con sus palabras qué es un foro.
- 2. ¿Conoce algún foro público? ¿Participa en alguno?
- 3. Dese de alta en algún foro que no sea de un compañero de clase. Haga capturas de todo el proceso de alta seguido.
- 4. Busque información sobre BbPress en páginas en español.
- 5. Localice en Internet algún otro gestor de foros, ya sea de pago o gratuito. Muestre las ventajas e inconvenientes de cada uno de los encontrados.



TEST DE CONOCIMIENTOS

- 1 BbPress aparece en el año:
 - a) 1980.
 - b) 1987.
 - c) 1995.
 - d) 2008.
 - e) 2011.
 - f) Todas son ciertas.
 - g) Todas son falsas.
- 2 Los antecesores de los foros son:
 - a) BBS.
 - b) Usenet.
 - c) Netuse.
 - d) SBB.
 - e) Las respuestas a y b son ciertas.
 - f) Las respuestas c y d son ciertas.
 - g) Todas son falsas.
- 3 BbPress es:
 - a) Gratuito.
 - b) Gratis los 3 primeros foros, a partir del cuarto son de pago.
 - c) De pago.
 - d) De pago solo para empresas, para los usuarios comunes es gratis.
 - e) Todas son falsas.
- 4 Para instalar bbPress necesito tener:
 - a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
 - b) Un servidor web con soporte PHP y MySQL.
 - c) Un equipo con Windows.
 - d) Un equipo con Linux.
 - e) Todas son ciertas.
 - f) Todas son falsas.
- 5 Para utilizar bbPress necesito tener:
 - a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
 - b) Un equipo normal con una conexión normal.
 - c) Un equipo con Windows.
 - d) Un equipo con Linux.
 - e) Todas son ciertas.
 - f) Todas son falsas.

9

Joomla!

OBJETIVOS DEL CAPÍTULO

- ✓ Definir qué es Joomla, historia de Joomla!.
- ✓ Diferenciar los componentes de Joomla!.
- ✓ Descargar e instalar Joomla!.
- ✓ Dar contenido a Joomla!.
- ✓ Conocer principales usuarios y papeles desarrollados.
- ✓ Conocer las plantillas.
- ✓ Realizar copias de seguridad de la información.

9.1 DEFINICIÓN

Joomla! es una pronunciación fonética de la palabra suajili *jumla*, que significa “todos juntos” o “como un conjunto”. Fue elegido para afirmar el compromiso total del equipo de desarrollo del núcleo Joomla! y los muchos colaboradores del proyecto Joomla! que trabajan juntos en este poderoso proyecto Open Source.

Joomla, como Software de Código Abierto (OSS) se puede descargar, instalar y modificar libremente y se puede utilizar sin restricciones, tanto para fines personales como comerciales. El código base de Joomla! está escrito en lenguaje PHP. La instalación de Joomla! puede realizarse en cuestión de minutos, en su ambiente nativo de desarrollo: Sistema Operativo Linux, Servidor Web Apache, Módulo PHP y Base de Datos MySQL (todos ellos considerados también entre las mejores aplicaciones de código abierto). Joomla! también puede instalarse en otros tipos de servidores, siempre que estén presentes el módulo PHP y la base de datos MySQL. La mayoría de los proveedores de *web hosting* cumplen los requisitos necesarios para la instalación de Joomla.

9.1.1 HISTORIA DEL PROYECTO

Joomla surge como el resultado de una bifurcación de Mambo, de la corporación Miro de Australia, quien mantenía la marca del nombre Mambo en esa época y el grupo principal de desarrolladores. Joomla nace con esta división el 17 de agosto de 2005. La corporación Miro formó una organización sin ánimo de lucro con el propósito inicial de fundar el proyecto y protegerlo de pleitos. El grupo de desarrollo reclamó que muchas de las cláusulas de la estructura de la fundación fueron acuerdos previos hechos por el comité directivo de Mambo, el cual no tiene la consultoría necesaria de quienes mantienen el proyecto e incluye cláusulas que violan los valores principales del código abierto. El grupo de desarrollo creó un sitio web que se llamó OpenSourceMatters para distribuir información a los usuarios, desarrolladores, diseñadores web y a la comunidad en general. En ese momento el líder Andrew Eddie, conocido como “MasterChief”, escribió una carta abierta para la comunidad, que apareció en la sección de anuncios del foro público en *mamboserver.com*.

Al siguiente día 1.000 personas ingresaron al sitio web *opensourcematters.org* para expresar su apoyo y estímulo por las acciones ejecutadas por el grupo de desarrollo. El sitio web recibió un aviso de temporalmente fuera de servicio debido al excesivo tráfico. Este evento apareció en *newsforge.com*, *eweek.com* y *ZDnet.com*. Peter Lamont CEO de Miro dio una respuesta en el artículo titulado *The Mambo Open Source Controversy - 20 Questions with Miro*.

El 22 de enero de 2008 se ha lanzado la versión 1.5 estable de Joomla, que incorpora notables mejoras en el área de seguridad, administración y cumplimiento con estándares W3C. Actualmente cuenta con una gran cantidad de componentes, módulos y *plugins*, aunque se pueden usar los de la versión anterior (1.0.X). Para esto, solo hay que activar un *plugin* incluido llamado System-Legacy que mejora considerablemente (no por completo) la compatibilidad con los mambots para Joomla! 1.0.X.

9.1.2 VERSIONES

La primera versión de Joomla! (Joomla! 1.0.0) fue publicada el 16 de septiembre de 2005 y se trataba de una evolución paralela (*fork*) mejorada de Mambo 4.5.2.3 combinada con modificaciones de seguridad y *antibugs*. Le siguió la versión 1.5 y Joomla! 1.6, mejorando considerablemente el sistema ACL de usuarios y la posibilidad de crear árbol de secciones ilimitado, evitando así las restricciones que tenía el antiguo sistema de clasificación de contenidos.

Tabla 9.1 Versiones de Joomla

Versión	Fecha de lanzamiento	Con soporte hasta
1.0	2005-09-16	2009-07-22
1.5 (LTS)	2008-01-22	2012-04-10
1.6	2011-01-10	2011-08-19
1.7	2011-07-19	2012-02-10
2.5 (LTS)	2012-01-24	2013-10-10
Color		Significado
Rojo		Versión sin soporte
Verde		Versión con soporte
Azul		Versión en desarrollo

9.2 LICENCIAS DE USO

El software Joomla CMS es de uso gratuito y de libre distribución (*OpenSource*), bajo licencia pública general (GNU/GLP).

Es bueno que sepa que es el propietario de su sitio Joomla pero no es propietario del sistema Joomla que utiliza para el mismo (en el sentido en que no podría venderlo como si fuese suyo). Esto quiere decir que hay ciertas leyes de *Copyright* que hay que respetar. Tenga en cuenta que el dinero que alguien paga por la construcción de un sitio utilizando Joomla no es por concepto de la compra del software (porque este es gratuito), sino que el cliente paga por los servicios que se derivan de Joomla: de instalación, configuración, diseño y puesta en marcha del sitio, además del respectivo soporte.

De todas formas, nadie nunca podrá “cobrar” por la utilización de Joomla o impedir que alguien utilice este sistema si no se violan los términos de la licencia GNU/GLP, además es necesario que sepa que el *Copyright* sobre los contenidos de cada sitio le pertenecen al propietario del mismo. Una cosa es el *copyright* de Joomla y otra muy distinta es el *copyright* de los contenidos de un determinado sitio.

9.3 REQUERIMIENTOS DE FUNCIONAMIENTO

Las características técnicas mínimas del servidor son:

- ✓ PHP 5.2.4 o superior (se recomienda PHP 5.3 o superior).
- ✓ MySQL 5.0.4 o superior.
- ✓ Apache 2.x o IIS 7.

9.4 DESCARGA DE JOOMLA

Los archivos principales de la distribución de Joomla (si los queremos en español) podemos descargarlos de <http://www.joomlaspanish.org/>. Una vez descargado el archivo, lo descomprimos y subimos los archivos a nuestro servidor web:



Figura 9.1. Descarga de Joomla

ACTIVIDADES 9.1



- Descargue y suba a su servidor web (puede ser local) la última versión de Joomla.

9.5 INSTALACIÓN DE JOOMLA

Suponemos que hemos subido a nuestro servidor los archivos de Joomla en una carpeta llamada joomla. Por lo tanto, el punto de partida será:

```
http://miservidor/joomla
```

Durante la instalación vamos a pasar por una serie de pantallas que nos van a requerir introducir unos datos, los pasos son los siguientes:

En la primera pantalla elegiremos el idioma, en nuestro caso he seleccionado **Español** (Figura 9.2). Una vez elegido el idioma se hace una comprobación del entorno para ver si cumple los requisitos mínimos para instalar Joomla. En caso de no cumplir estos requisitos no podremos pasar de este apartado (Figura 9.3).

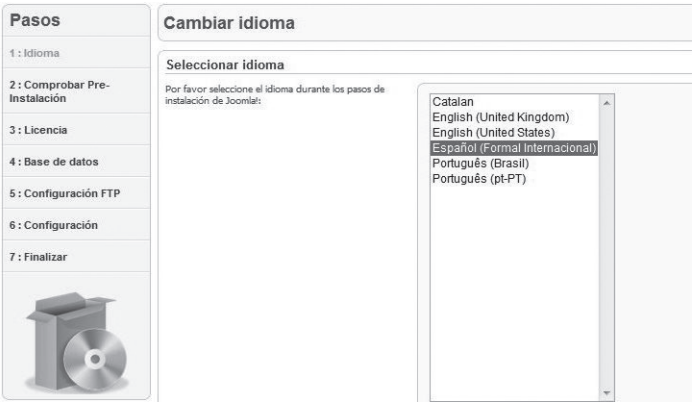


Figura 9.2. Selección de idioma



Figura 9.3. Comprobación de requisitos

En la siguiente pantalla nos aparece la licencia GNU de Joomla (Figura 9.4). Después configuramos la base de datos: aquí nos recomienda usar MySQL. En este apartado debemos indicar el *host* del servidor de bases de datos, el usuario y la contraseña. También debemos asignar un nombre a la base de datos de Joomla y el prefijo para las tablas de la base de datos para Joomla (Figura 9.5).

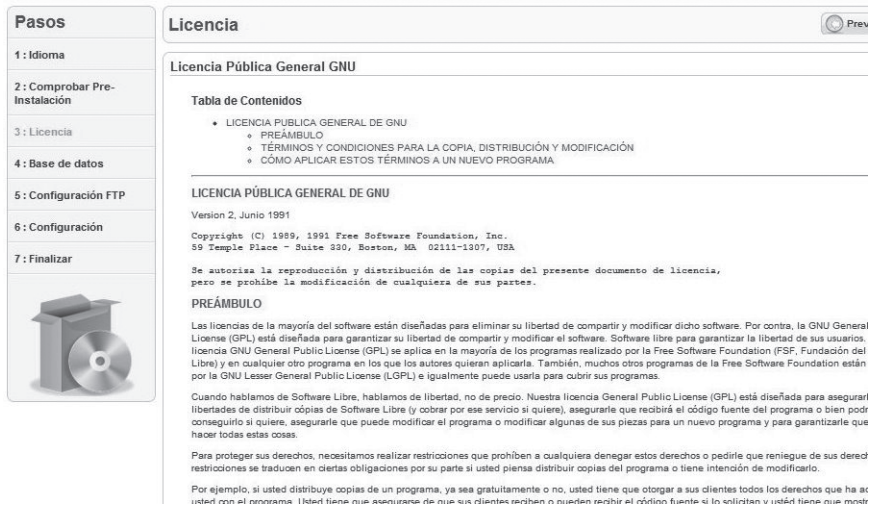


Figura 9.4. Licencia de Joomla

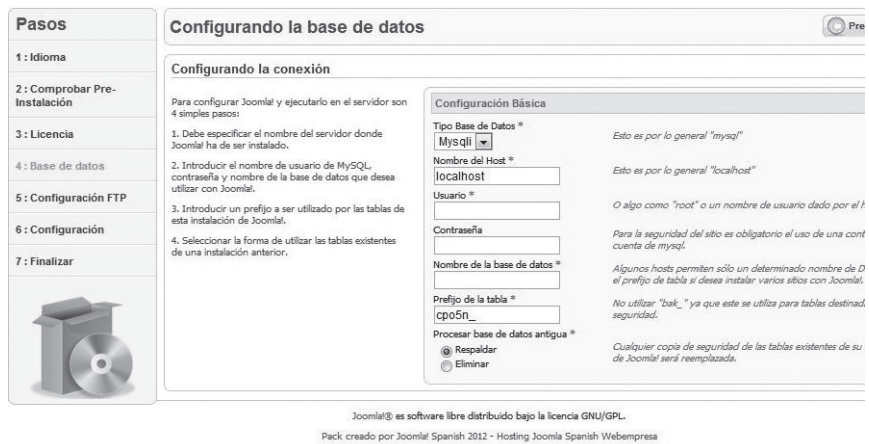


Figura 9.5. Configuración de la BD

A continuación aparece la configuración del FTP del servidor. Habrá que configurar este apartado en caso de que nuestro servidor así lo requiera (Figura 9.6).

Pasos

- 1: Idioma
- 2: Comprobar Pre-Instalación
- 3: Licencia
- 4: Base de datos
- 5: Configuración FTP
- 6: Configuración
- 7: Finalizar

Configuración FTP (Opcional - La mayoría de usuarios puede omitir este paso) Pulse siguiente para continuar

FTP Configuración (Opcional)

En algunos servidores es posible que deba proporcionar las credenciales FTP para que la instalación sea completada. Si tiene dificultades para completar la instalación sin estas credenciales, consulte con su Hosting para determinar si esto es necesario.

Por razones de seguridad, lo mejor es crear una nueva cuenta de usuario FTP con acceso a la instalación de Joomla! y no al servidor web completo. Su Hosting le puede ayudar con esto.

Nota: Si está instalando en un sistema operativo de Windows, la capa FTP no es necesaria.

Configuración Básica

Habilitar Capa FTP *

Usuario FTP

Contraseña FTP

Ruta Raíz FTP

[Configuración Avanzada](#)

Figura 9.6. Configuración FTP

Tras esta pantalla insertaremos la configuración principal: nombre del sitio, *email* y datos de la cuenta de administrador. También podemos instalar los datos de ejemplo (recomendado para usuarios noveles, como nosotros) (Figura 9.7).

Pasos

- 1: Idioma
- 2: Comprobar Pre-Instalación
- 3: Licencia
- 4: Base de datos
- 5: Configuración FTP
- 6: Configuración
- 7: Finalizar

Configuración principal

Nombre del sitio

Introduzca un nombre para su sitio web Joomla!.

Configuración Básica

Nombre del Sitio *

[Configuración avanzada - Opcional](#)

Confirmar la contraseña y correo electrónico del usuario admin.

Introduzca su dirección de e-mail. Esta será la dirección de correo electrónico del sitio Web vinculada al Super Administrador.

Introduzca una nueva contraseña y luego confírmela en los campos correspondientes. Usted puede cambiar el nombre de usuario por defecto **admin**. Estos serán los nombres de usuario y contraseña que utilizará para acceder al Panel de Control de administrador al final de la instalación.

Si va a migrar, puede ignorar esta sección. Su configuración actual se migrará automáticamente.

Cargar datos de ejemplo (Español)

Importante! Es altamente recomendable a los nuevos usuarios de Joomla! instalar los datos de ejemplo por defecto. Para ello haga clic en el botón antes de pasar a la siguiente fase.

Antes de salir de la instalación, puede rellenar la base de datos del sitio con los datos. Hay varias maneras de hacer esto:

1a. Los datos por defecto se pueden insertar. Para ello haga clic en el botón Instalar Datos de Ejemplo.

1b. Los datos de ejemplo personalizados pueden ser introducidos. Para ello, seleccione una muestra de datos disponibles en el menú desplegable y haga clic en el botón Instalar los datos.

Datos de ejemplo

La instalación de los datos de ejemplo es muy recomendable para principiantes. Esto instalará el contenido de ejemplo que se incluye en el paquete de Joomla!.

Figura 9.7. Configuración del sitio

Ya hemos terminado la instalación, tan solo tenemos que eliminar la carpeta de instalación y ya podremos disfrutar de nuestro Joomla (Figura 9.8). Para acceder al área de administración de nuestro Joomla debemos pulsar sobre **Administración** y nos aparecerá una pantalla de autenticación como la de la Figura 9.9.



Figura 9.8. Finalizar



Figura 9.9. Acceso a administración

Para ver cómo ha quedado nuestra nueva página Joomla debemos introducir en el navegador la dirección *http://miservidor/joomla*. Ya podremos ver los resultados de la instalación.

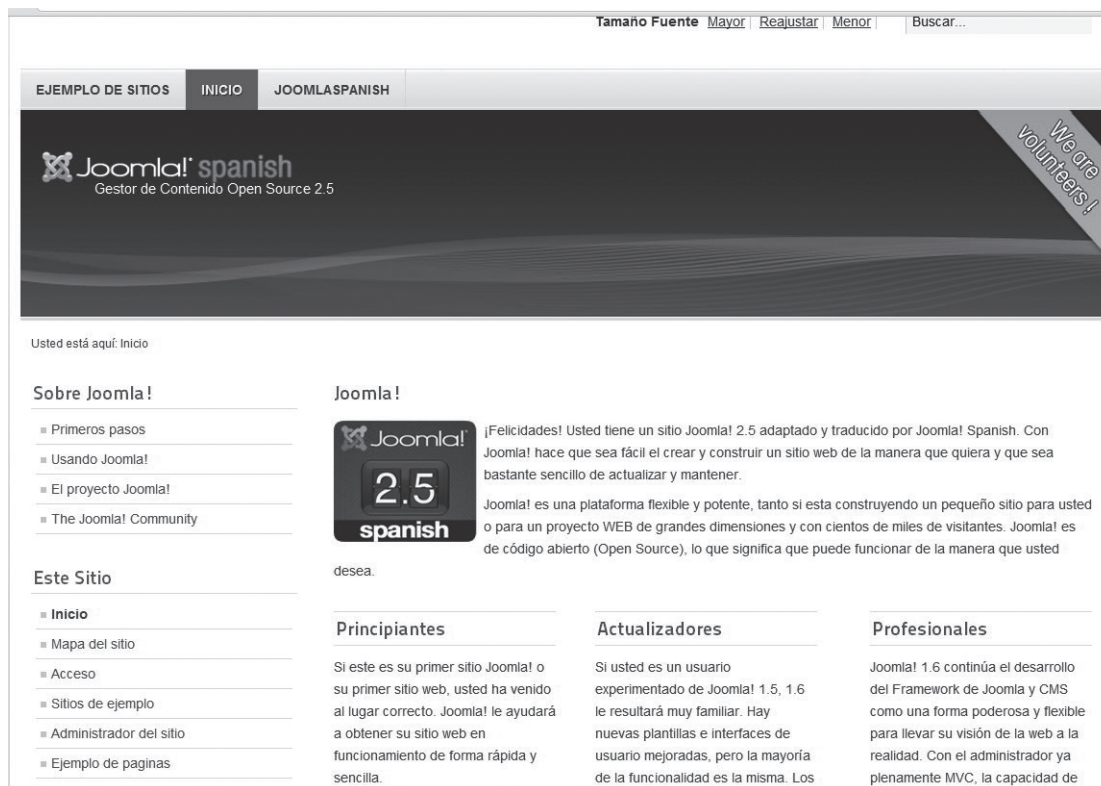


Figura 9.10. Joomla

ACTIVIDADES 9.2



- Instale Joomla en su servidor web.

9.6 ESTRUCTURA DE JOOMLA

En Joomla es importante organizar el contenido mediante una estructura jerárquica: categorías, subcategorías y artículos (noticias, enlaces, etc), lo que constituirá el orden interno del CMS. En los siguientes apartados vamos a ver cómo definir esta estructura.

9.7 CATEGORÍAS

Joomla es un enorme archivador que contiene cajones (categorías) que, a su vez, contienen carpetas (subcategorías), que son los que guardan los artículos. No es posible publicar un artículo exclusivamente en una categoría. Deberemos seleccionar una categoría y su subcategoría para publicar un artículo. Una categoría puede tener una o varias subcategorías y una subcategoría podrá contener uno o más artículos.

Para crear o editar categorías, desde el **Panel de control** hacemos clic en el siguiente icono (Figura 9.11):



Figura 9.11. Gestor de Categorías

Aquí podemos crear nuevas categorías, editar existentes, publicar y despublicar categorías existentes, etc. (Figura 9.12).

Gestor Categorías: Artículos

Nuevo

Editar

Publicar

Despublicado

Archivo

Comprobar

Papelera

Reconstruir

Opciones

Ayuda

Artículos

Categorías

Artículos destacados

Filtro:

Buscar

Limpiar

- Seleccione Max Niveles -

- Seleccione Estado -

- Seleccionar Acceso -

- Seleccione Idioma -

	Título	Estatus	Orden	Acceso	Idioma	ID
	Datos ejemplo Artículos (Alias: sample-data-articles)		1	Publico	Todo	14
	Joomla! (Alias: joomla)		1	Publico	Todo	19
	Extensiones (Alias: extensiones)		1	Publico	Todo	20

Figura 9.12. Categorías

Si pulsamos sobre **Crear categoría** deberemos rellenar los datos que nos pide para crear la nueva categoría. En nuestro caso vamos a crear una nueva categoría llamada *Hardware* (que tendrá una serie de subcategorías) (Figura 9.13). Ahora ya tenemos nuestra nueva categoría que hemos creado (Figura 9.14).

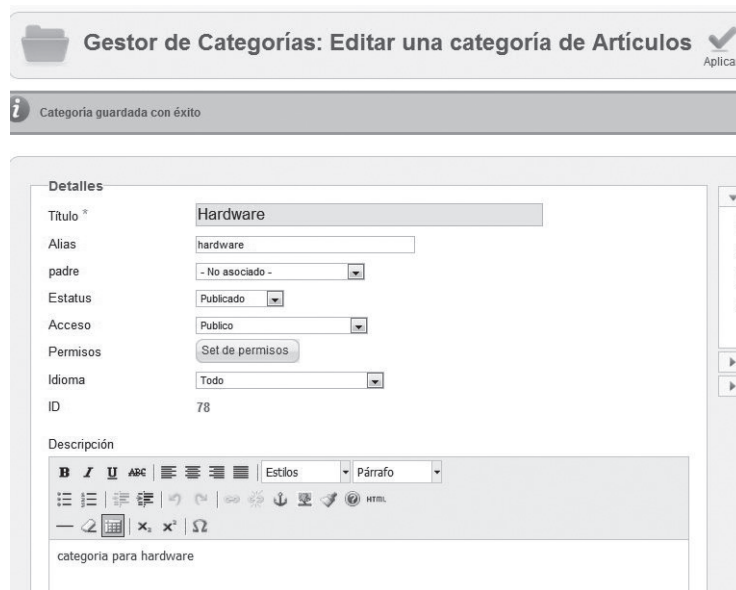


Figura 9.13. *Nuevar categoría*

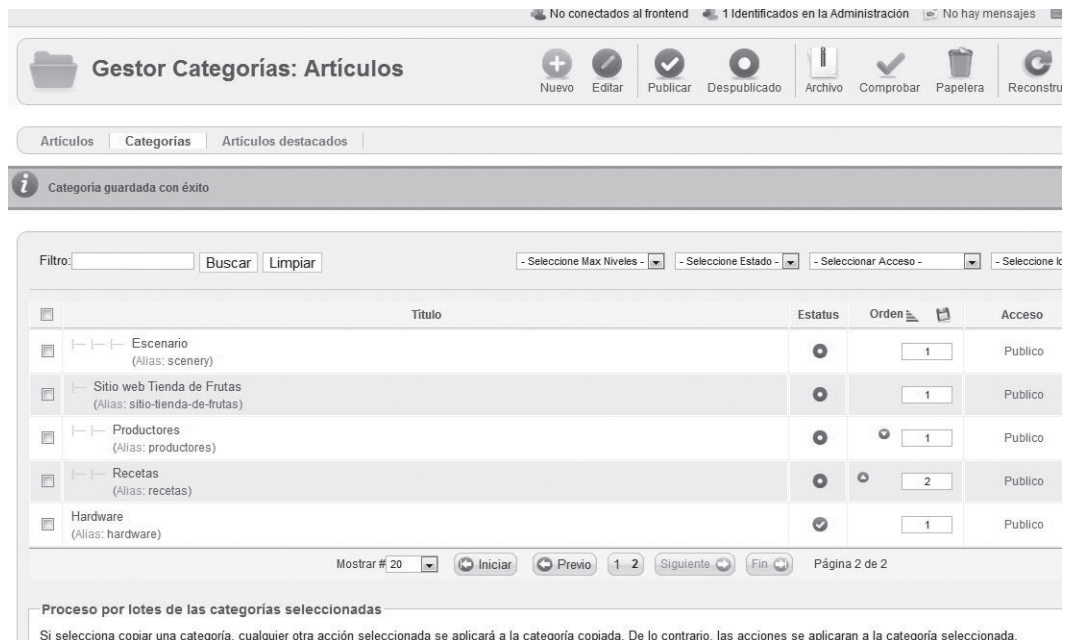


Figura 9.14. Categoría Hardware

Vamos a crear una subcategoría nueva llamada *Procesadores* que va a depender de la categoría *Hardware* (Figura 9.15).

Gestor de Categorías: Añadir nueva categoría de Artículos ✓ Aplicar Guardar

Detalles

Título *

Alias

padre

Estatus

Acceso

Permisos

Idioma

ID

Descripción

B I U ABC

Opciones de publicación

Creado por

Opciones básicas

Opciones Metadatos

Figura 9.15. SubCategoría

Podemos ver como *Procesadores* es una subcategoría de *Hardware* (Figura 9.16).

☐	☐	Productores (Alias: productores)	☐	1	Publico	Todo
☐	☐	Recetas (Alias: recetas)	☐	2	Publico	Todo
☐	☒	Hardware (Alias: hardware)	☒	1	Publico	Todo
☐	☐	Procesadores (Alias: procesadores)	☒	1	Publico	Todo

Mostrar # 20 1 2 Página 2 de 2

Proceso por lotes de las categorías seleccionadas

Si selecciona copiar una categoría, cualquier otra acción seleccionada se aplicará a la categoría copiada. De lo contrario, las acciones se aplicaran a la categoría seleccionada.

Establecer el nivel de acceso

Figura 9.16. Subcategoría Procesadores

ACTIVIDADES 9.3



➤ Cree una categoría nueva y tres subcategorías que pertenezcan a esta nueva categoría creada.

9.8 PLANTILLAS

Una de las ventajas de Joomla es la facilidad con la que podemos cambiar su aspecto. Existen múltiples plantillas para instalar en Joomla, la mayoría de ellas de forma gratuita. Además, podemos instalar cuantas plantillas deseemos sin modificar el contenido de la Web, ya que el diseño y el contenido son independientes.

Las plantillas web son muy importantes, ya que determinarán la posición de los diferentes módulos que habrá en la página web. Es por ello que la selección de una plantilla adecuada para nuestra *website* sea ciertamente muy importante.

Podemos gestionar las plantillas instaladas desde la Administración de Joomla. Hay dos tipos de plantillas: unas son para el **sitio** (lo que el cliente web va a visualizar) y otras son para la **administración** (solo se utilizará esta plantilla en el área de administración).



Figura 9.17. Plantillas

Podemos descargar cientos de plantillas para sitio desde la web <http://www.joomla24.com> (y otras muchas). Vamos a ver cómo descargar plantillas de Joomla, instalarlas y fijarlas en nuestro portal.

Nos vamos a la página que hemos nombrado anteriormente: <http://www.joomla24.com> (Figura 9.18).

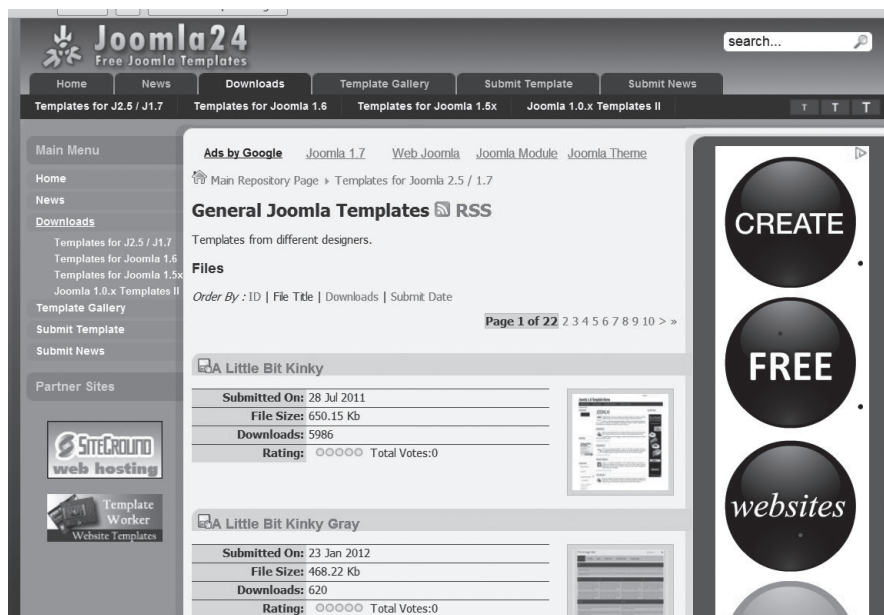


Figura 9.18. www.joomla24.com

Pulsamos sobre **Downloads** y elegimos nuestra versión de Joomla. Nos aparecerán muchas plantillas, podemos echarles un vistazo y elegir la que consideremos más adecuada a nuestra futura web (Figura 9.19).

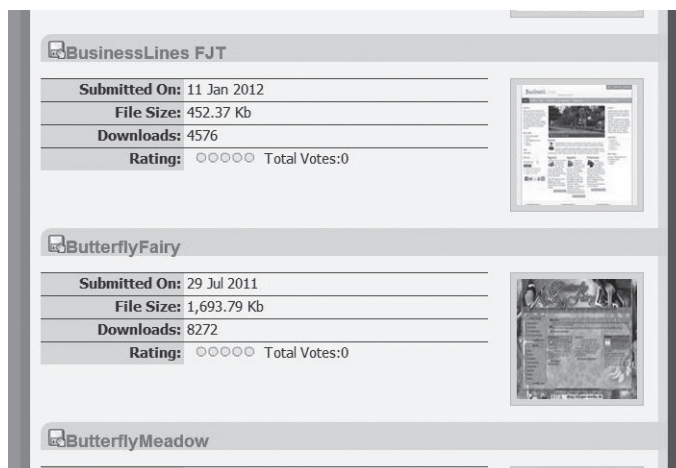


Figura 9.19. Downloads

Una vez descargada la plantilla que nos guste. Nos vamos al área de administración de Joomla, a **Gestor de Extensiones**. Seleccionamos el archivo que hemos descargado y le damos a **Subir e instalar** (Figura 9.20).

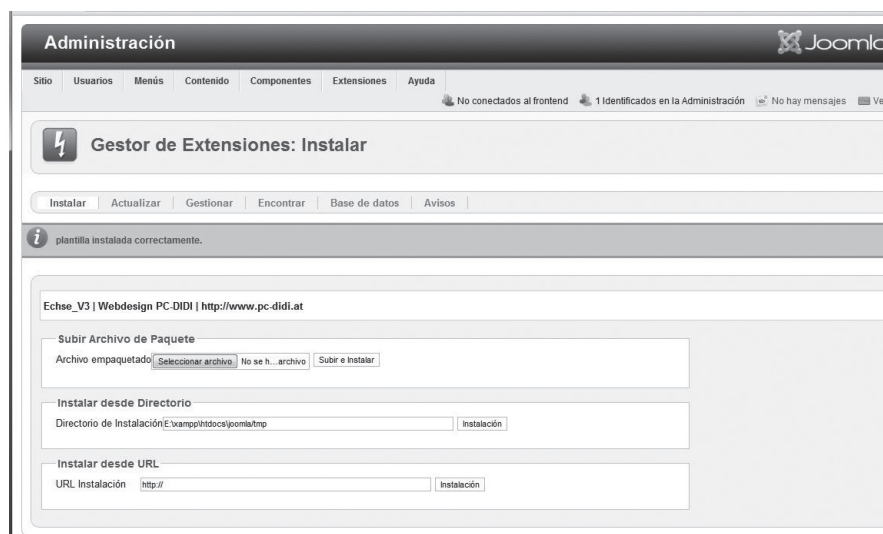


Figura 9.20. Instalar extensión

Una vez hemos instalado la nueva plantilla nos vamos a **Gestor de plantillas** y allí, entre otras, tenemos la plantilla que hemos descargado e instalado. Pulsamos sobre el cuadro que hay antes del nombre de la plantilla que acabamos de instalar y pulsamos en **Establecer por defecto** (Figura 9.21).

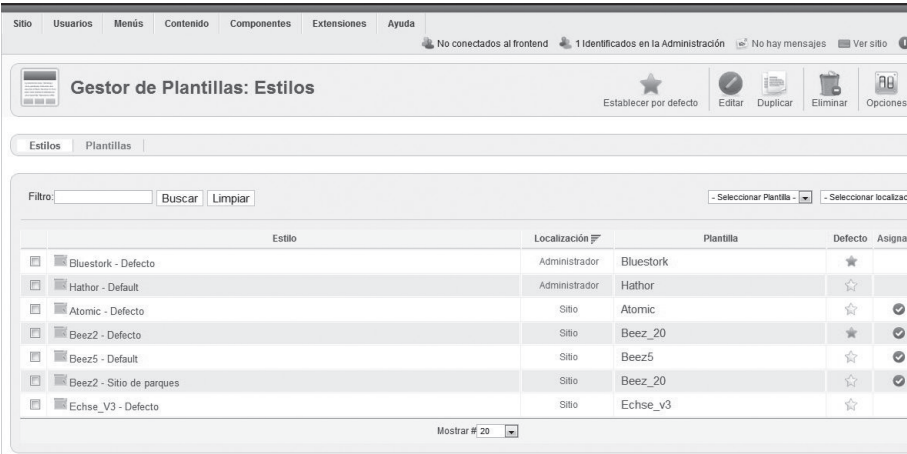


Figura 9.21. Establecer por defecto

Ya podemos ver nuestra nueva web resultado de la plantilla anterior (Figura 9.22). Podemos tener tantas plantillas como queramos instaladas en el sistema, aunque solo podremos tener una activa en un momento dado.



Figura 9.22. Nueva plantilla

ACTIVIDADES 9.4



- Vaya a la web www.joomla24.com, descargue e instale una plantilla en su sistema y establézcala como predeterminada.

9.9 USUARIOS Y GRUPOS

En Joomla podemos distinguir dos zonas, una zona pública (la web en sí) denominada **frontend** y otra restringida a cada uno de los diferentes administradores, llamada **backend**.

Los visitantes podrán registrarse en la página web y acceder de esta forma a los distintos recursos “no publicados”, limitados a estos usuarios. Así, por ejemplo, estos usuarios podrán enviar noticias, enlaces, etc. De esta forma, los administradores de la web deciden que contenidos están limitados a usuarios registrados y cuáles no.

Los usuarios que deseen registrarse, lo podrán hacer en cuanto el módulo **Registro** esté activado, rellenando un simple cuestionario. Una vez realizado el proceso de registro, podrán loguearse en la web, mediante el nombre de usuario y contraseña.

Los niveles de permisos están inteligentemente asignados, ya que cuando un usuario ingresa, el sistema le provee acceso en el menú solo a los recursos autorizados y no hay forma de que se “salte” esta seguridad para acceder a aquellos recursos no autorizados o que requieren un nivel de permisos mayor.

9.9.1 NIVELES DE ACCESO A LA PARTE PÚBLICA (FRONTEND)

Los usuarios que se han identificado en el *frontend* solo pueden acceder a esta zona, desde aquí no podrán acceder a la zona de administración (*backend*).

- **Registrado.** Un usuario registrado puede acceder a la información que requiere de este nivel de autenticación, pudiendo acceder a las descargas y en general a cualquier recurso que el administrador esté con el nivel de acceso **registered**.
- **Autor.** Puede enviar artículos, pero no puede editar los ítems de los demás, ni publicar los que escriba él mismo (quedan pendientes validación hasta que un publicador o administrador lo autorice y publique).
- **Editor.** Puede enviar artículos y editar los de los demás, pero no puede publicar (estos quedan **Pendientes de validación**, al igual que el caso anterior).
- **Publicador.** Puede enviar, modificar y publicar ítems.

9.9.2 NIVELES DE ACCESO A LA PARTE DE ADMINISTRACIÓN (BACKEND)

- **Superusuario.** Posee acceso total a todo el sitio y a toda la administración
- **Administrador.** Similares atributos al del superusuario pero sin la posibilidad de ingresar en la configuración global. Tampoco puede editar o eliminar al superadministrador.
- **Gestor.** Tiene un acceso parcial en el *admin*, sin poder acceder a la administración de componentes, a los módulos, a la edición de usuarios, tampoco a la configuración, etc.

9.9.3 CREAR O ADMINISTRAR USUARIOS EN JOOMLA

El usuario **superusuario** es el único con capacidad total de manipular usuarios, registros y permisos. Inclusive puede denegarse a sí mismo la entrada al sitio. Los usuarios **registrados** son los únicos capaces de crearse a sí mismos con el solo hecho de registrarse en el sitio, siempre y cuando el administrador permita el registro de usuarios. El superusuario puede elegir entre permitir el registro de usuarios, o puede guardarse esa capacidad solo para él,

creando usuarios registrados desde el administrador. Para esto solo configura un parámetro dentro de la configuración global *Allow user registration*.



Figura 9.23. Usuarios

Desde el panel de control del sitio, nos vamos a **Gestor de Usuarios**.

Ahí, pulsamos en **Añadir** o en **Modificar** (según lo que queramos hacer). En nuestro caso, elegimos **Añadir un nuevo usuario**, rellenamos los datos del usuario y le asignamos un grupo al que va a pertenecer (Figura 9.24).

Figura 9.24. Usuarios

Una vez completos los datos y asignado el grupo pulsamos en **Aplicar**. Nuestro nuevo usuario ha sido creado (Figura 9.25).



Figura 9.25. Usuarios

ACTIVIDADES 9.5



- Cree tres usuarios nuevos que pertenezcan cada uno a un nivel distinto de acceso.

9.10 MENÚS

Los menús son módulos, hay distintos tipos de menú y pueden ser creados tantos menús como necesitemos. Inclusive pueden mostrarse distintos menús dependiendo de la sección o página por la que navegue el visitante. Los menús que vienen por defecto en Joomla son:

- **Menú principal** (*mainmenu*): es el menú principal del sitio y el más importante. Lo que se encuentre como primer elemento de este menú es lo que se mostrará en la página principal.
- **Menú de tipo Top**: este es el menú horizontal que aparece sobre (o debajo) de la cabecera del sitio.
- **Menú de usuario** (*usermenu*): solo aparece cuando un usuario se registra e ingresa en el sistema.
- **Otros menús**: es simplemente “otro menú” cualquiera.

En el siguiente ejemplo vamos a ver cómo añadir un menú más a los enlaces (Figura 9.26).



Figura 9.26. Enlaces

Nos vamos a **Gestor de Menús** (Figura 9.27).



Figura 9.27. Gestor de Menús

Dentro de **Menús**, seleccionamos **Ejemplos Enlaces**, que es el que contiene los menús de enlaces al que queremos añadir el nuevo elemento (Figura 9.28).

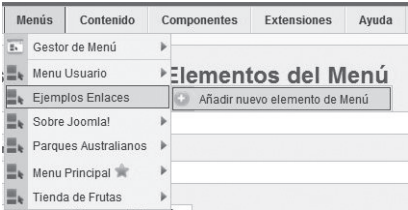


Figura 9.28. Ejemplos Enlaces

Añadimos el nuevo menú llamado **Hardware** (Figura 9.29).

Figura 9.29. Menús

Ya tenemos el nuevo enlace (Figura 9.30).



Figura 9.30. Enlace nuevo

ACTIVIDADES 9.6



- Cree dos menús nuevos, uno que sea de tipo *menú de enlace* (como el que hemos visto en el apartado) y otro de tipo *mainmenu*.

9.11 MÓDULOS Y COMPONENTES

La principal página de descargas de estas extensiones es <http://extensions.joomla.org/>. Vamos a mostrar cómo instalar un módulo que nos va a mostrar la hora en nuestra web. Hacemos clic en **Calendar & events, Time, Clocks** (Figura 9.31).

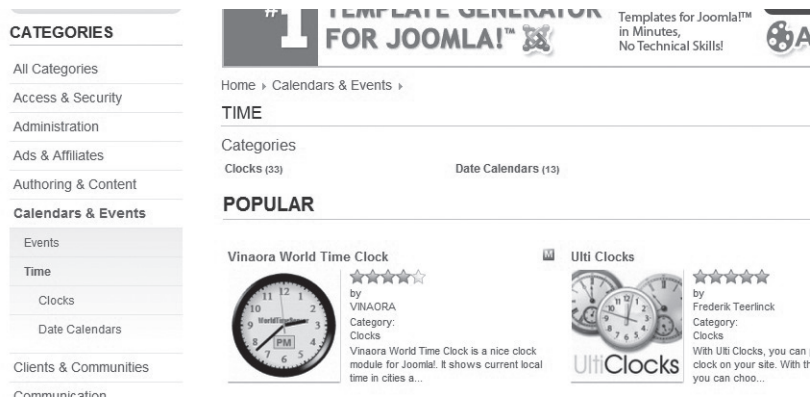


Figura 9.31. extensions.joomla.org

Pulsamos sobre uno de los relojes (el que creamos conveniente para nuestra web). Dentro de la página de características muestra las versiones con las que es compatible y podemos descargarlo (Figura 9.32).

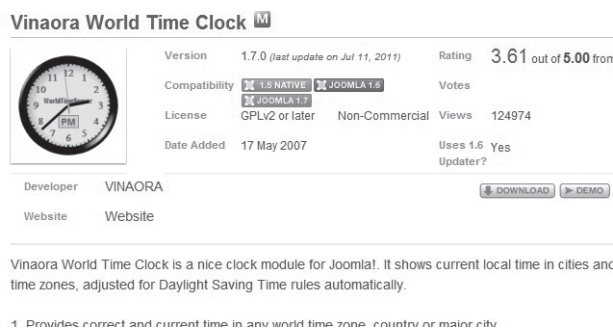


Figura 9.32. Relojes

Si no estamos en la zona de administración de Joomla nos dirigimos a ella. Dentro de esta zona nos dirigimos al gestor de extensiones (Figura 9.33).



Figura 9.33. Gestor de extensiones

Seleccionamos el archivo y le damos a **Subir e instalar** (Figura 9.34).



Figura 9.34. Instalar extensión

Ahora nos vamos a **Gestor de módulos**. Podemos ir desde el Panel de control o desde el menú **Extensiones**, **Gestor de módulos** (Figura 9.35). Aquí nos aparece la nueva extensión (Figura 9.36).



Figura 9.35. Gestor de módulos

	Título	Estatus	Posición
☐	Vinaora World Time Clock		Ninguno
☐	Módulo de...		...

Figura 9.36. Nueva extensión

Pulsamos sobre ella para editarla, aquí podemos seleccionar la posición donde irá en la web, si queremos que aparezca publicado, las páginas donde aparecerá, etc. Una vez asignadas las opciones pulsamos en **Aplicar** para que los cambios tengan efecto y podamos ver el resultado (si hemos seleccionado **Publicar**, claro está) (Figura 9.37).

Ya tenemos nuestro nuevo y maravilloso módulo, un reloj que ilumina nuestra web. Si hacemos una vista previa de la página podemos observar el nuevo módulo dentro de ella (Figura 9.38).

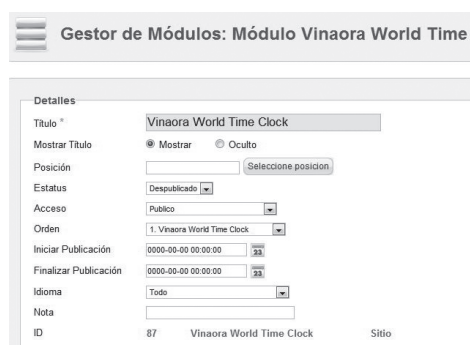


Figura 9.37. Edición nueva extensión



Figura 9.38. Vista previa del reloj

ACTIVIDADES 9.7



- Añadir a nuestro Joomla una extensión llamada **Phoca Gallery** que pertenece a la sección **Photos & Images**. Verá que después de instalarlo aparece un menú especial para él dentro de *Componentes*. Hacer capturas de todo el proceso y del nuevo menú aparecido.

9.12 ARTÍCULOS

Desde el **Gestor de Artículos** de Joomla! los administradores/as pueden modificar, añadir o eliminar los artículos de contenido disponibles en su sitio web. Joomla!. Permite además gestionar por separado los artículos publicados en la página de inicio y los artículos archivados (Figura 9.39).



Figura 9.39. Gestor de artículos

En esta pantalla veremos dos zonas diferenciadas: la Barra de herramientas en la zona superior y la Lista de artículos en el área central de trabajo (Figura 9.40).

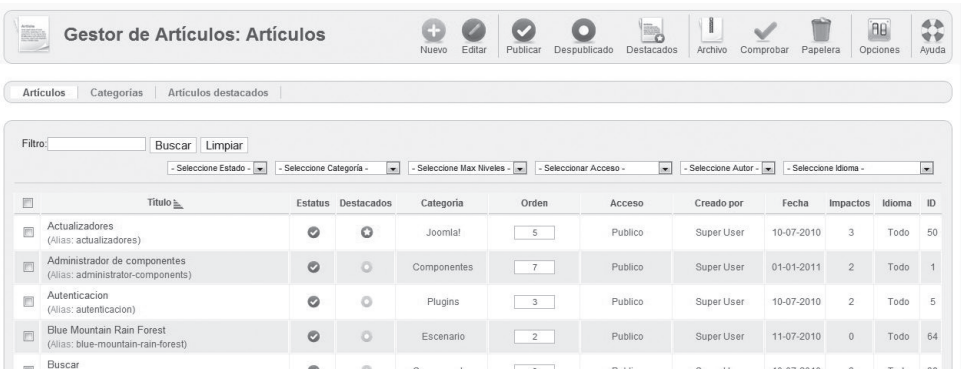


Figura 9.40. Gestión de artículos

9.12.1 LISTA DE ARTÍCULOS

La lista de artículos publicados en el sitio se muestra dentro de una tabla con varias columnas (Figura 9.41).

	Título	Estatus	Destacados	Categoría	Orden	Acceso	Creado por	Fec
☐	Actualizadores (Alias: actualizadores)	☒	☐	Joomla!	5	Publico	Super User	10-07
☐	Administrador de componentes (Alias: administrator-components)	☒	☐	Componentes	7	Publico	Super User	01-01
☐	Autenticación (Alias: autenticaion)	☒	☐	Plugins	3	Publico	Super User	10-07
☐	Blue Mountain Rain Forest (Alias: blue-mountain-rain-forest)	☒	☐	Escenario	2	Publico	Super User	11-07
Buscar								

Figura 9.41. Lista de artículos

Se puede filtrar la lista de artículos introduciendo un texto de búsqueda en el campo *Filtro* (Figura 9.42).

Figura 9.42. Filtro

También se pueden filtrar los artículos por sección, categoría, autor o estado seleccionando una o más combinaciones en las listas desplegables. Además, pulsando sobre el encabezado de las columnas podremos ordenar la lista según ese criterio.

Nos aparece un listado con los artículos que tenemos en nuestra web. En dicho listado se muestra toda la información necesaria para administrarlos. En primer lugar, nos aparece el nombre del artículo con un enlace que si lo pulsamos nos llevará a la zona de edición de dicho artículo. A continuación nos aparece un icono de estado, que nos informa de si nuestro artículo está publicado o no. Si pulsamos en **Estatus** podemos publicarlo o despublicarlo. El icono **Destacados** funciona de forma similar, es decir, al pulsarlo vamos cambiando a los distintos estados permitidos.

Figura 9.43. Estatus y destacados

9.12.2 EDITAR ARTÍCULOS

Si queremos editar un artículo, tan solo tenemos que pulsar sobre el nombre y nos aparecerá dicho artículo para ser modificado (Figura 9.44).

Figura 9.44. Editar artículo

9.12.3 NUEVO ARTÍCULO

Antes de insertar un nuevo artículo debemos tener creadas las categorías y subcategorías a las que pertenecerá dicho artículo. Desde el panel de administración nos aparece el icono **Añadir un nuevo artículo** (Figura 9.45).



Figura 9.45. Añadir nuevo artículo

Debemos poner un nombre al artículo y un alias, establecer a qué categoría pertenece, si va a estar publicado, etc.

El formulario se divide en varias secciones. En la parte superior, hay un encabezado con el título "Gestor de Artículos: Añadir Nuevo Artículo" y botones "Aplicar" y "Guardar". A la izquierda, se encuentran los campos de entrada para el artículo: "Titulo" (con el valor "Nuevo procesador i14"), "Alias" (con el valor "i14"), "Categoría" (con un menú desplegable que muestra "-- Procesadores"), "Estatus" (con un menú desplegable que muestra "Publicado"), "Acceso" (con un menú desplegable que muestra "Publico"), "Permisos" (con un botón "Set de permisos"), "Destacados" (con un menú desplegable que muestra "No") y "Idioma" (con un menú desplegable que muestra "Todo"). Debajo de estos campos, hay un campo "ID" con el valor "0". En la parte inferior izquierda, hay un editor de texto con una barra de herramientas que incluye botones para negrita, cursiva, subrayado, lista, etc. En la parte superior derecha, hay un menú con opciones como "Opciones de Publicación", "Opciones Artículo", "Configurar la pantalla", "Imágenes y enlaces" y "Opciones Metadatos".

Figura 9.46. Añadir nuevo artículo

ACTIVIDADES 9.8



- Edite dos artículos de los existentes y añada otros dos nuevos a su página. Los nuevos enlázelos a dos categorías diferentes.

9.13 COPIAS DE SEGURIDAD

Para las copias de seguridad vamos a utilizar otra extensión de Joomla llamada **LazyDbBackup**. Esta extensión, además de ser gratuita y tener licencia GPL, realiza *backups* automáticos con la periodicidad que le indiquemos. Además, si configuramos el correo, nos enviará un *email* con dicha copia de seguridad.

Partimos de <http://extensions.joomla.org/>. Nos dirigimos a la sección **Site Security** y dentro de ésta a **Backup** (copia de seguridad) (Figura 9.47).

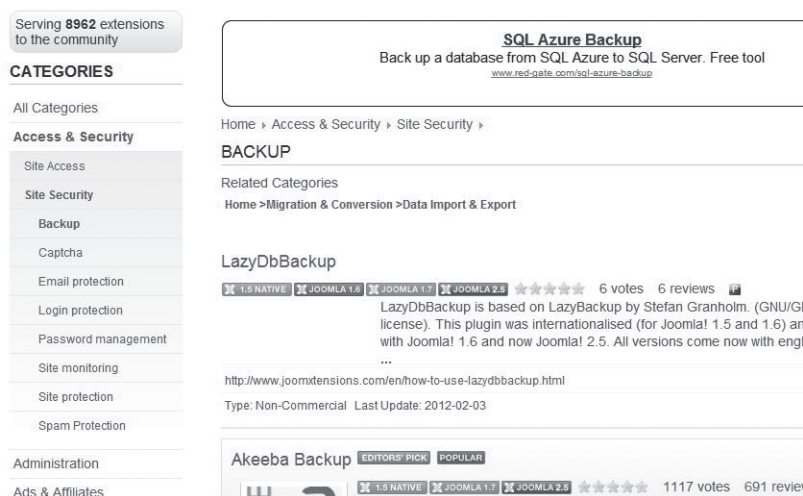


Figura 9.47. *extensions.joomla.org*

Seleccionamos **LazyDbBackup** (o el que más nos guste), lo descargamos y lo instalamos desde el gestor de extensiones (Figura 9.48).



Figura 9.48. *Gestor de extensiones*

Una vez instalado nos vamos al gestor de *plugins* (Figura 9.49). Dentro del gestor de *plugins* buscamos ahí la extensión instalada **LazyDbBackup** (Figura 9.50).

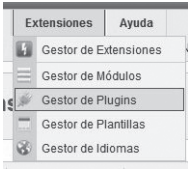


Figura 9.49. Gestor de *plugins*

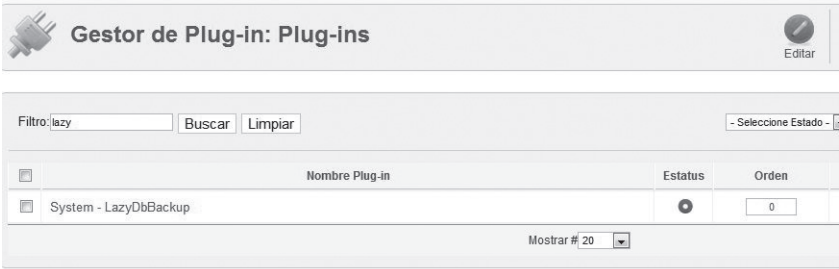


Figura 9.50. LazyDbBackup

Lo primero que debemos hacer es activarlo, ya que si no está activado no se realizará la copia de seguridad (Figura 9.51). En la parte derecha aparece la configuración. En este apartado podemos configurar las tablas que hacer *backup*, la periodicidad de la copia, el *email* donde será enviada, etc. (Figura 9.52).

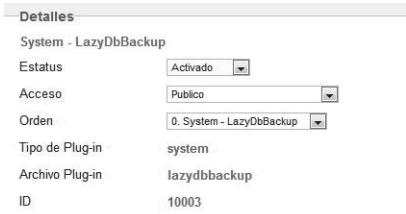


Figura 9.51. Acitvar LazyDbBackup

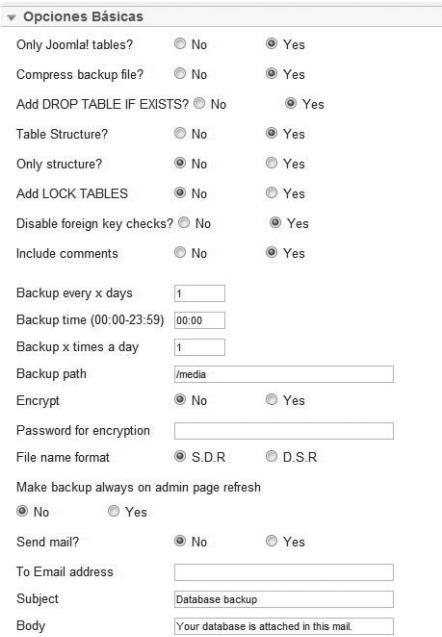


Figura 9.52. Configuración de LazyDbBackup

ACTIVIDADES 9.9



- Instale la extensión para copias de seguridad **LazyDbBackup** en su sistema. Programe las copias para que se hagan cada dos días y que se le envíe un correo con la copia a su dirección de *email*.

9.14 ACTUALIZACIONES

Al entrar al área de administración, Joomla comprueba automáticamente si existen actualizaciones y nos las muestra por pantalla.



Figura 9.53. Actualizaciones



Figura 9.54. Actualizaciones



Antes de efectuar cualquier actualización es recomendable realizar una copia de seguridad de la base de datos de Joomla.

ACTIVIDADES 9.10



- Haga una captura del nivel de actualizaciones de su sistema.



RESUMEN DEL CAPÍTULO

Este es el último capítulo dedicado a los gestores de contenidos. Nos hemos dejado para este último a la estrella de los CMS: Joomla.

Joomla incluye una serie de características como son: mejorar el rendimiento web, permitir versiones imprimibles de páginas, incorporar *flash* con noticias, *blogs*, foros, *polls* (encuestas), calendarios, búsqueda en el sitio web y un largo etcétera.

Lo que se ha presentado en este capítulo ha sido una pequeña introducción a todo este mundo (ya que si se quisiese tratar a fondo Joomla necesitaríamos varios libros como éste). Ánimo al estudiante y lector a ahondar en este tema, ya que hay multitud de sitios web dedicados a Joomla, tanto con tutoriales como con recursos.



EJERCICIOS PROPUESTOS

- **1.** Defina con sus palabras qué es Joomla y para qué sirve.
- **2.** Buscar en la Web páginas que estén hechas con Joomla.
- **3.** Buscar información sobre Joomla en páginas en español.
- **4.** Localice las principales diferencias entre las versiones 1.X de Joomla y la nueva 2.5.
- **5.** Hay versiones de Joomla que llevan la etiqueta LTS. Investigue qué significa esa etiqueta. Busque otros productos que también posean la etiqueta LTS.



TEST DE CONOCIMIENTOS

- 1** La primera versión de Joomla aparece en el año:
 - a) 1992.
 - b) 1997.
 - c) 2005.
 - d) 2008.
 - e) 2011.
 - f) Todas son ciertas.
 - g) Todas son falsas.
- 2** El antecesor de Joomla es:
 - a) BBS.
 - b) Usenet.
 - c) Mambo.
 - d) Lajoom.
 - e) Las respuestas a y b son ciertas.
 - f) Las respuestas c y d son ciertas.
 - g) Todas son falsas.
- 3** Joomla es:
 - a) Gratuito.
 - b) Gratis el primer portal, a partir del segundo son de pago.
 - c) De pago.
 - d) De pago solo para empresas, para los usuarios comunes es gratis.
 - e) Todas son falsas.
- 4** Para instalar Joomla necesito tener:
 - a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
 - b) Un servidor web con soporte PHP y MySQL.
 - c) Un equipo con Windows.
 - d) Un equipo con Linux.
 - e) Todas son ciertas.
 - f) Todas son falsas.
- 5** Para utilizar Joomla necesito tener:
 - a) El mejor equipo que haya en el mercado con la conexión más rápida existente.
 - b) Un equipo normal con una conexión normal.
 - c) Un equipo con Windows.
 - d) Un equipo con Linux.
 - e) Todas son ciertas.
 - f) Todas son falsas.

10

Implantación de aplicaciones de ofimática web

OBJETIVOS DEL CAPÍTULO

- ✓ Definir qué son las aplicaciones ofimáticas web.
- ✓ Conocer las aplicaciones de ofimática web más importantes.
- ✓ Utilizar una de las aplicaciones webs más utilizadas en la actualidad: Google Docs.

10.1 TIPOS DE APLICACIONES

Las aplicaciones web ofimáticas son una alternativa gratuita para editar textos, crear hojas de cálculo o realizar presentaciones sin necesidad de instalar uno de los programas tradicionales de escritorio como el Office de Microsoft. Se manejan a través de un navegador y solo requieren una conexión a Internet. Reúnen además numerosas ventajas para colaborar con otros usuarios en la edición simultánea de documentos, algo impensable hasta la fecha, ya que habilitan numerosas y variadas opciones para desarrollar los más diversos trabajos en equipo.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a muchos usuarios.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones.

10.1.1 VENTAJAS

- **Ahorra tiempo:** se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- **No hay problemas de compatibilidad:** basta tener un navegador actualizado para poder utilizarlas.
- **Espacio:** no ocupan espacio en nuestro disco duro.
- **Actualizaciones inmediatas:** como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- **Consumo de recursos bajo:** dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- **Multiplataforma:** se pueden usar desde cualquier sistema operativo porque solo es necesario tener un navegador.
- **Portables:** es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (solo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- **Disponibilidad:** suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- **Virus:** no dañan los datos porque estos están guardados en el servidor de la aplicación.
- **Colaboración:** gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones *on line* de calendarios u oficina.
- **Funcionalidades:** los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas (RIA).

10.1.2 INCONVENIENTES

- Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo. Pero cada vez los navegadores están más preparados para mejorar en este aspecto. La aparición de HTML 5 representa un hito en este sentido.
- La disponibilidad depende de un tercero, el proveedor de la conexión a Internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.

10.1.3 PROGRAMAS MÁS POPULARES

Este tipo de programas pueden servir de reemplazo a las suites ofimáticas convencionales, como Microsoft Office u Open Office. Los tres servicios más populares, todos ellos gratuitos para el usuario son Google Docs, Zoho y Thinkfree Online.

Los tres se caracterizan por su sencillez de uso, sobre todo Google Docs que, como de costumbre en los productos del famoso buscador, apuesta por potenciar la usabilidad y por un diseño muy limpio. La curva de aprendizaje respecto a las aplicaciones que se instalan en el escritorio es mínima. Thinkfree es muy similar a Microsoft Office tradicional (el anterior a la versión de 2007) y los botones cumplen las mismas funciones.

Los tres servicios coinciden en ofrecer tres aplicaciones básicas: el procesador de textos, la hoja de cálculo y el software de presentaciones. Google añade la posibilidad de enviar formularios, útiles para realizar invitaciones y confirmaciones de asistencia, mientras que Thinkfree dispone del extra de un editor WYSWYG que permite insertar en sus documentos diferentes contenidos de Internet, como fotos de Flickr o vídeos de YouTube. Sin embargo, Zoho sobresale en este aspecto con sus 12 aplicaciones distintas, entre ellas una wiki, un programa de correo electrónico, un calendario o un software para controlar otros ordenadores a distancia, con vistas a realizar un soporte técnico.

Estos programas no resultan tan completos como los que se instalan en el ordenador. Thinkfree proporciona más opciones de edición que el resto y Google Docs es el más básico. La colaboración de varias personas conectadas por Internet para trabajar en un mismo documento funciona bien en Google Docs y en Zoho, aunque siempre que se usa durante un período prolongado el programa funciona con lentitud.

Google Docs destaca por su mayor flexibilidad en este aspecto y hasta 10 personas pueden editar a la vez un texto, incluso dentro de un mismo párrafo. Sin embargo, Zoho ofrece un sistema menos ágil que bloquea el párrafo modificado para que no se solapen los cambios. Thinkfree se desmarca de sus dos rivales porque no contempla la edición simultánea y solo se puede trabajar en los contenidos de forma individual.

En general, mientras los procesadores de texto trabajan con una agilidad similar a la de las aplicaciones de escritorio, las hojas de cálculo son más lentas y los programas de presentaciones proporcionan menos opciones (con Thinkfree y Zoho como los mejores) respecto al software correspondiente que se instala en el ordenador. Por otro lado, Zoho y Thinkfree ofrecen 1 gigabyte para almacenar todos los documentos, una capacidad mayor que el límite de 5.000 archivos de Google Docs. La propuesta del buscador también queda la última de la prueba en cuanto al tamaño de los contenidos que se pueden cargar en el servidor. Zoho y Thinkfree han establecido un límite de 10 megabytes, mientras que Google Docs solo iguala esta cifra cuando se trata de presentaciones y reduce el tamaño hasta los 500 kilobytes para textos y un megabyte para las hojas de cálculo. Google Docs y Zoho trabajan con los principales formatos ofimáticos, pero Thinkfree se olvida del soporte al OpenDocument, el estándar libre que utiliza Open Office. Los tres servicios exportan a PDF.

Este tipo de servicios ha incorporado una nueva prestación gracias a las aplicaciones ricas de Internet. Se puede trabajar, por tanto, sobre los archivos aunque no se tenga conexión a Internet y sincronizarlos con la versión del

servidor más tarde. Así, es posible iniciar un viaje en tren, autobús o avión y trabajar sin conexión durante el trayecto para recuperarla cuando haya finalizado, momento en que se sincronizarán las dos versiones y se validará la más reciente. Zoho fue el primero en incorporar esta opción, paradójicamente con Gears, una tecnología de Google. Docs lo añadió en 2009, mientras que Thinkfree utiliza un programa convencional para desarrollar esta tarea.

Con el fin de crear documentos en colaboración, Google Docs es la mejor opción por su mayor facilidad de uso, mientras que Zoho es una apuesta segura para los interesados en disponer de diversas aplicaciones en un solo sitio destacando, además, por su hoja de cálculo. Por su parte, Thinkfree atrae a quienes desean trabajar en un sistema similar al Office tradicional, aunque la utilización que hace de Java origina que sea más lento en tareas como abrir o guardar los contenidos cuando se trabaja en conexión a la Red.

Tabla 10.1 Comparativa de programas más populares

	Google Docs	Zoho	Thinkfree
Soportan Office 2007	Sí	Solo el .docx	Solo el .docx
Soportan ODF	Sí	Sí	No
Exportan a PDF	Sí	Sí	Sí
Número de programas	4	12 (en su versión personal)	4
Edición en grupo simultánea	Sí	Sí	No
Capacidad del documento que se puede subir	Documentos: 500 KB; Presentaciones 10 MB; Hojas de cálculo 1 MB, PDF 10 MB	10 MB	10 MB
Enviar documentos por correo electrónico	Sí (también creación)	Sí	Sí
Número de tipos de letra	11 (Verdana por defecto)	24 (Verdana por defecto)	Las del equipo
Capacidad de almacenamiento	5.000 documentos y 5.000 imágenes	1 GB	1 GB

10.1.4 GOOGLE DOCS

- **Lo mejor:** facilidad de uso. Buscador interno dentro de los documentos. Corrector ortográfico.
- **Lo peor:** no exporta bien la alineación justificada. Menor tamaño de los documentos que se pueden importar. Limitación de los tipos de letra.
- ✓ **Aplicaciones:** procesador de textos, hoja de cálculo, presentaciones y formularios.

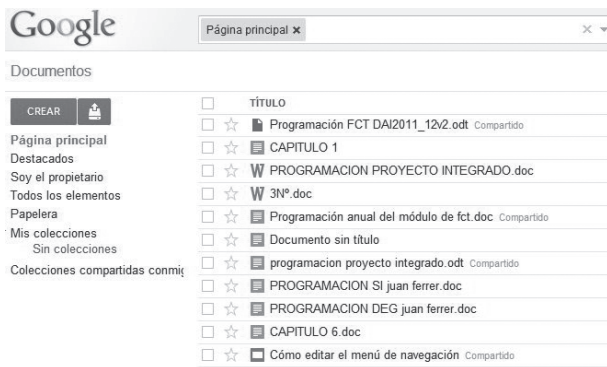


Figura 10.1. Google Docs

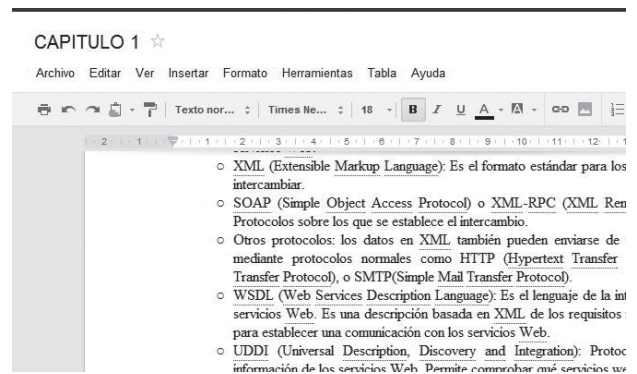


Figura 10.2. Google Docs

- ✓ **Características y manejo.** El gran punto a favor de Google Docs se halla en su diseño minimalista y su usabilidad. Es sencillo acostumbrarse a esta aplicación y un gran número de sus acciones se hacen intuitivas para quienes manejan de manera habitual programas ofimáticos. Además, está traducido al español.

Esta aplicación permite trabajar con los archivos ofimáticos más extendidos, como los de Microsoft Office y los estándares libres de OpenDocument. Exporta a HTML, RTF, CSV y a PDF. Igual que Zoho y Thinkfree, Google Docs permite compartir las tareas de creación en dos niveles, uno para invitar a colaboradores y otro solo a lectores. Hasta 200 personas pueden tener permiso para editar los documentos de texto y las presentaciones, aunque de ellos solo 10 pueden trabajar en un mismo documento a la vez. En el caso de las hojas de cálculo, el límite de editores asciende hasta 50 de forma simultánea.

Google posibilita la publicación de los contenidos en modo página web con añadidos como un RSS para que otras personas puedan informarse de las novedades. Destaca la opción de traspasar la propiedad de un documento para que, a partir de ese momento, otra persona se convierta en la dueña del mismo. Así mismo, integra un programa para enviar por correo electrónico formularios con distintos tipos de respuestas. Por medio de éste se pueden idear invitaciones, encuestas o confirmar la asistencia a un acto, por ejemplo.

En la página de inicio de Google Docs, los contenidos se clasifican por carpetas gracias a la utilización de Ajax. Destaca el cuadro de búsqueda que rastrea los propios archivos guardados en el servidor con el fin de encontrar los términos deseados. Por otro lado, ofrece plantillas para diferentes documentos de texto, hojas de cálculo y presentaciones. Una prestación de gran utilidad para crear tarjetas de visita, listas de invitados o llevar la contabilidad doméstica, entre otras tareas.

Siempre que la labor se centra en colaborar en la edición de un documento, la aplicación avisa de que otra persona está realizando ya esta tarea, y lo hace mediante un pequeño letrero que se superimpone en la parte inferior derecha de la página. En el procesador de textos mantiene sin bloquear la zona del texto que se edita para que otros colaboradores puedan realizar sus modificaciones. Respecto a la hoja de cálculo, muestra con colores las celdas donde otro usuario escribe. Los editores solo pueden *chatear* cuando trabajan sobre una hoja de cálculo, una labor útil para coordinarse en equipo. Esta opción no la ofrece ni el procesador de textos ni el software de presentaciones, a diferencia de Zoho, que potencia esta forma de comunicación en sus aplicaciones. Cada documento cuenta con su propia página donde constan todos los cambios efectuados, denominada **Historial de revisiones**.

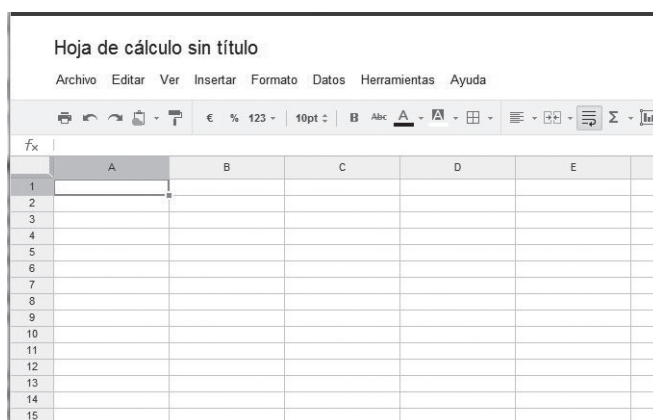


Figura 10.3. Google Docs

- ✓ **Las aplicaciones básicas.** En el procesador de textos destaca su corrector ortográfico, que personaliza su diccionario con los términos que designe el usuario y sugiere alternativas a los posibles fallos. El sistema de búsqueda resalta de forma automática las palabras con un fondo amarillo.

La creación automática de un texto por medio de un correo electrónico (a través de la extracción de la información del cuerpo del mensaje y el asunto) funciona con corrección y respeta los estilos que se hayan añadido en la redacción del mensaje, como las negritas o los enlaces. Sin embargo, el paso inmediato de un adjunto del correo electrónico al programa sin descargarlo es muy lento.

Dispone de 11 tipos de letra, el dato más bajo de los obtenidos en la prueba práctica, con un cuerpo a elegir entre 8 y 36 puntos. Sorprende que no se encuentre la alineación justificada entre sus botones de acceso directo. Se ha comprobado que este tipo de alineación causa problemas cuando se exporta un documento a una aplicación de escritorio (la última línea del párrafo se ensancha en lugar de ocupar la anchura correspondiente).

En las hojas de cálculo se pueden añadir gráficos o *gadgets* (pequeñas aplicaciones que muestran los datos de forma diferente). Por otro lado, es útil la creación de nombres de rango que agrupan varias celdas para simplificar las diferentes fórmulas. Se trata de una de las características más importantes ya que hasta ahora solo estaba disponible en las hojas de cálculo de sobremesa, las más avanzadas.

Las presentaciones disponen de cinco estructuras predefinidas y fondos de distintos colores. También se pueden insertar imágenes (desde la Web y desde el ordenador del usuario) o vídeos desde YouTube gracias a un buscador integrado. Al igual que Zoho, facilita invitaciones para que otros usuarios puedan disfrutar de la presentación, permitiendo *chatear* con los mismos a la vez.

- ✓ **Uso sin Internet:** el trabajo sin conexión funciona a través de Gears, la aplicación de Internet realizada por Google. Se instala con rapidez a partir del enlace **Sin conexión**, que se encuentra en la página central de todos los documentos. A Google Docs se puede acceder sin conexión cuando se introduce la dirección <http://docs.google.com> en el navegador y permite la lectura de las presentaciones y hojas de cálculo, aunque solo se puede editar el texto. Con el fin de mantener la sincronización de los datos, hay que conectarse de nuevo a la Red. Solo así el servidor de Google Docs actualizará la última versión.

10.1.5 ZOHOO

- **Lo mejor:** es el servicio que más aplicaciones ofrece, el más compatible con formatos ofimáticos. Posibilita *chatear* en los principales programas. Destaca su hoja de cálculo.
- **Lo peor:** traducción al español incompleta. Su diseño es barroco.
- ✓ **Aplicaciones:** Procesador de textos, Hoja de cálculo, Presentaciones, Correo, Calendario (agenda), Wiki, Gestor de tareas, Notas, Enlaces, Libreta de contactos, Creador de libros y Reunión.

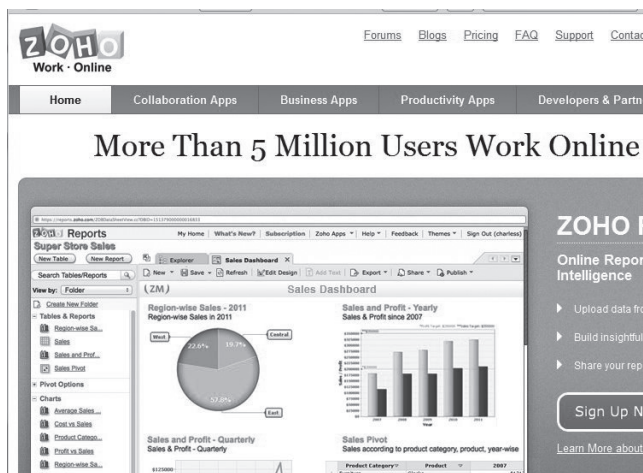


Figura 10.4. Zoho

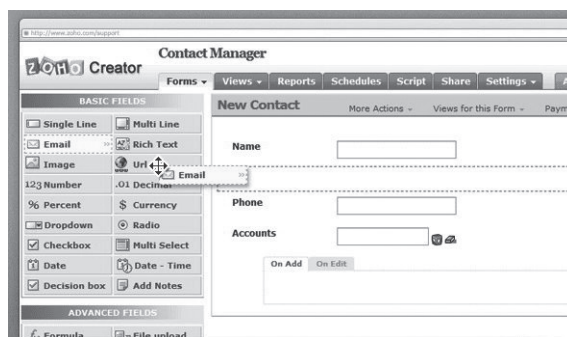


Figura 10.5. Zoho

- ✓ **Características y manejo.** Zoho se caracteriza por el gran número de aplicaciones gratuitas que pone a disposición de los usuarios, 12 en su configuración personal más el añadido de otras dirigidas al público empresarial. Cuenta con una wiki, un programa de correo electrónico, un calendario y un programa para tomar el control de otro ordenador a distancia, de forma que se pueda ofrecer soporte o ayuda técnica. Cifra que supera la oferta de Google Docs y Thinkfree. Estas prestaciones están traducidas al español, aunque de una forma incompleta, con algunas palabras en inglés.

Zoho brilla por su compatibilidad con los formatos ofimáticos. Además de trabajar con los del Microsoft Office anteriores a 2007, como Google Docs y Thinkfree, también soporta el *.docx*, el sucesor del *.doc* en la versión 2007, los documentos a OpenDocument, el estándar libre; LaTeX, un formato indicado para la publicación de libros de forma profesional; SXW, el formato original del Open Office, PDF, RTF, TXTi, CSV, TSV, XHTML y HTML.

Al igual que Google Docs, centraliza todas las aplicaciones en una página, denominada Zoho Personal, aunque no permite acceder de un solo vistazo a todos los archivos sin importar el formato, una configuración que le resta agilidad.

En su programación utiliza Ajax, gracias a lo cual el usuario puede modificar la apariencia con acciones como variar la anchura de las columnas, arrastrar archivos o mover bloques del diseño. Dentro de Zoho Personal se puede acceder al Escritorio, que muestra un resumen de los últimos archivos guardados con las aplicaciones, de los últimos correos electrónicos recibidos y de las próximas tareas recopiladas en la agenda. También se puede acceder a Documentos, donde se permite la gestión de todos los archivos creados que, en total, pueden llegar a ocupar 1 gigabyte. Solo en una aplicación utiliza Java (en Reunión) ideada para citarse con otras personas con el fin de conversar o para ayudarlas desde el punto de vista técnico porque asume en remoto el control de los equipos.

Zoho facilita la publicación de contenidos en la Web, de igual manera que Google Docs. Aunque el historial de revisiones no resulta tan intuitivo como en Google Docs, cumple su cometido: muestra las personas que editan el documento en ese mismo momento. Su principal ventaja es que permite *chatear* mientras se edita no solo en el procesador de textos, sino también en la hoja de cálculo o en el software de presentaciones.

Además, calca la configuración de los *chats* tradicionales y posibilita el envío de mensajes en privado o al canal público si hay varios editores. Sin embargo, la velocidad en las labores de edición es respecto a Google Docs. La razón es que cuando una persona hace cambios en un documento se bloquea el párrafo o contenido en cuestión para que otro no borre los cambios realizados en ese preciso instante.

Por otra parte, ofrece más prestaciones que el resto de los servicios analizados en el manejo del programa en móviles o a través de complementos para el navegador. Dispone de una versión para el iPhone del procesador de textos, la hoja de cálculo y el software de presentaciones y otra para los terminales con sistema operativo Windows Mobile. Por otro lado, incluye un complemento, Zoho Plugin (instalado en el Microsoft Office para guardar los textos y las hojas de cálculo en Zoho) con el fin de obtener una copia de seguridad o favorecer la colaboración en grupo.

Las aplicaciones básicas

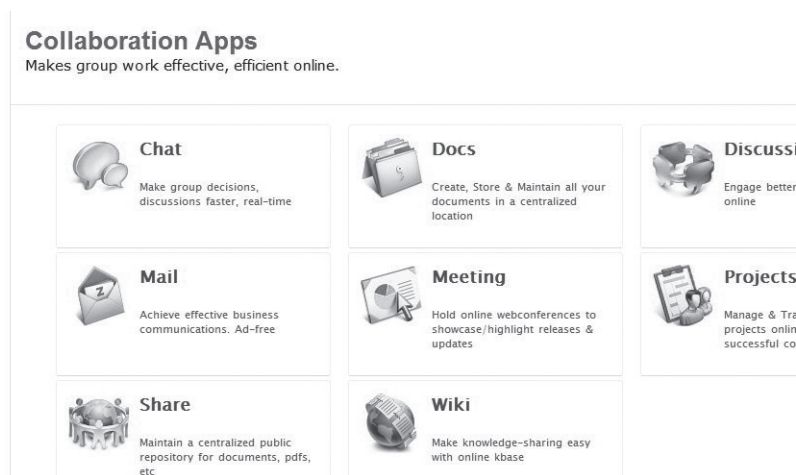


Figura 10.6. Zoho

El procesador de textos no está diseñado con la sencillez del de Google Docs y los iconos se han colocado de manera desordenada. Por otro lado, es el servicio con mayor cantidad de tipos de letra, 24, con un tamaño que va de los 8 puntos a los 36 y cuyas prestaciones son más numerosas que en Google Docs.

Se agradece la opción de **Page Setup**, situada en la parte superior de la página (su utilidad es la de poner un logotipo o el número de página, por ejemplo) y la parte inferior (para los datos de contacto).

Al igual que en la aplicación del popular buscador, permite insertar una hoja de estilo CSS. Se puede configurar también el espacio entre líneas, una característica con la que no cuenta Google Docs. La exportación del documento para trabajar después en un programa tradicional funciona bien y los estilos aplicados se trasladan sin problemas.

La hoja de cálculo de Zoho reúne más prestaciones que sus rivales; puede realizar macros, vincularse con datos externos a partir de RSS o CSV, crear tablas pivote o dar un formato condicional a las tablas. El programa de presentaciones proporciona más de 50 plantillas prediseñadas, de fácil acceso, a diferencia de Google Docs, donde

deben buscarse en una página aparte. Además, cuenta con una surtida variedad de formas, *clip arts*, flechas, leyendas o estrellas. Como en el caso de Google Docs, se puede enseñar la presentación mientras se chatea con otras personas. Sin embargo, no permite introducir vídeos dentro de la misma.

- ✓ **Uso sin Internet:** el procesador de textos de Zoho es la única aplicación junto con el correo electrónico que puede funcionar sin conexión a Internet gracias a Gears. Un enlace en la parte superior, denominado **Go offline**, permite pasar a este modo y descargar al ordenador un determinado número de documentos a elección del usuario en los que se puede trabajar sin necesidad de conectarse a Internet. Para diferenciar el modo, la parte superior de la pantalla se torna gris. Cuando se regresa a la Red, Zoho informa de las versiones modificadas y avisa al usuario de que se van a sobrescribir con la versión local, una opción que se puede rechazar.

10.1.6 THINKFREE

- **Lo mejor:** su aspecto casi idéntico a Microsofts Office. Numerosas prestaciones para la edición de textos.
- **Lo peor:** lentitud en la carga de Java. Imposibilidad de que trabaje más de una persona a la vez en la edición de un documento.
- ✓ **Aplicaciones:** Procesador de textos, Hoja de cálculo, Presentaciones, Notas (para incluir elementos de la web en documentos).

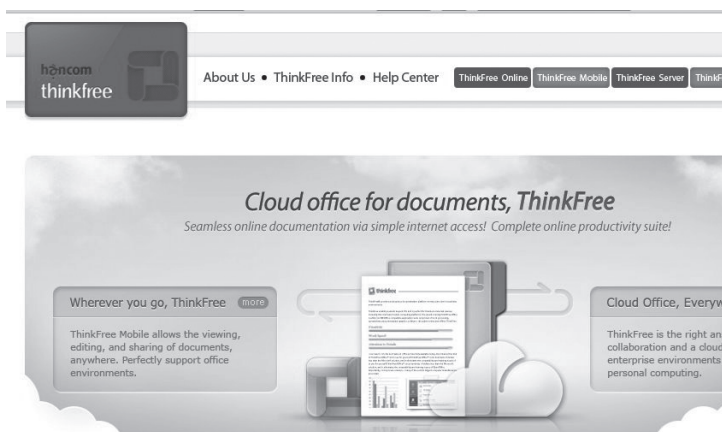


Figura 10.7. ThinkFree

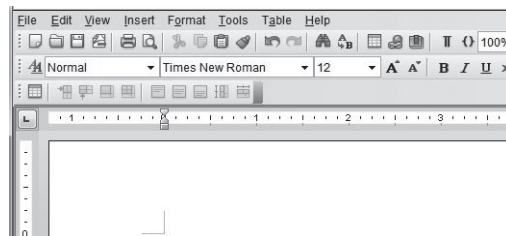


Figura 10.8. ThinkFree

- ✓ **Características y manejo:** Thinkfree es un caso singular dentro de las aplicaciones web ofimáticas porque no permite la edición simultánea, como sí sucede en el caso de Google Docs o Zoho. Esta característica obliga a realizar una edición individual, ya que no deja trabajar a varias personas sobre un documento, por lo que le resta agilidad al proceso. Un inconveniente que se agudiza porque ni siquiera se facilita el envío del archivo a otra persona, tarea que exige salir del editor. Tampoco dispone de un sistema de *chat*, innecesario según este planteamiento.

Los documentos se comparten de manera distinta porque el sistema crea un enlace a una página web para visualizar el documento y luego brinda la posibilidad a los que no son usuarios de descargarlo para su edición en una aplicación ofimática tradicional y de editarlo a los que sí lo son.

La gran apuesta de Thinkfree consiste en que estéticamente se asemeja bastante al aspecto de Microsoft Office hasta el lanzamiento de la versión 2007 y al actual diseño de Open Office. Así, en su manejo logra transmitir la sensación de que se utiliza una aplicación instalada en el ordenador. Ahora bien, esta impresión se frustra por los retrasos que ocasiona el uso de Java cada vez que se abre o se guarda un documento. Si se desea escribir un texto hay que esperar alrededor de un minuto mientras Thinkfree carga el Java de la ventana emergente. Durante la prueba de uso, Java ha dado errores y no se ha podido abrir el documento deseado, un inconveniente que se soluciona borrando la caché de Java y reiniciando el ordenador. Otra desventaja radica en que no está traducido al español, aunque sí cuenta con un corrector ortográfico en nuestra lengua.

Al igual que Google Docs, se centra en las aplicaciones básicas de la ofimática y como extra ofrece **Note**, un editor WYSIWYG, que permite escribir en *blogs* y sitios web.

Su tarea consiste en simplificar el añadido de vídeos de YouTube, fotos de Flickr y mapas de Yahoo! en combinación con la creación de los contenidos ofimáticos de esta *suite*. Por otro lado, como Google Docs y Zoho, ofrece la posibilidad de publicar los documentos en otros sitios web.

Tal y como caracteriza a Zoho, no exige que los usuarios se registren en su sitio, y pueden utilizar su servicio directamente todas las personas que se hayan abierto una cuenta de Google. Thinkfree soporta el formato de texto del Office 2007 (.docx), los del Office tradicional, más RTF, TXT, PDF, XML y HTML. Es sorprendente que no permita almacenar en ODF, el estándar de Open Document aprobado por la ISO, a diferencia de Google Docs y Zoho. Sin embargo, es de gran utilidad que permita guardar de forma directa los archivos al ordenador. De esta forma, no se necesita exportarlos como sucede con Google Docs o Zoho, sino que se puede trabajar de manera directa, aunque el programa funcione desde el navegador.

Las aplicaciones básicas

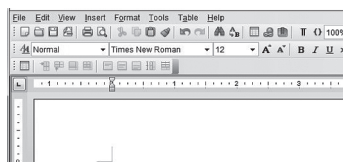


Figura 10.9. ThinkFree

El procesador de textos dispone de una gran variedad de tipos de letra porque los toma del propio ordenador, y el tamaño de los mismos es el más amplio de todos los de la prueba de uso, desde 8 hasta 72 puntos.

Cuando se trata de teclear las palabras o introducir estilos, el programa funciona muy bien, aunque se demora en guardar los documentos. Por sus características, es completo y permite realizar formatos de texto avanzados, con más opciones de edición que sus rivales.

Por otra parte, las hojas de cálculo se sitúan en un nivel intermedio en cuanto a prestaciones entre Zoho, la mejor opción en este campo, y Google Docs. En las presentaciones cuenta con mayor número de opciones que los otros dos servicios; se pueden incrustar vídeos (AVI, MPEG, MOV), imágenes de Flickr o del ordenador, archivos de sonido MP3, WAV o MIDI y añadir animaciones personalizadas a cada diapositiva.

- ✓ **Uso sin Internet:** Thinkfree también se diferencia de la competencia en su uso sin conexión a la Red. Opta por un programa bastante pesado (más de 40 megabytes) que logra agilizar el funcionamiento de la versión en línea y facilitar la sincronización de documentos cuando se trabaja desconectado. A diferencia de la versión en línea, que solo se encuentra en inglés, este programa sí incluye la traducción al español. Además, se puede instalar en Windows, Mac OS X y Linux.

10.2 GOOGLE DOCS

Como se ha visto en el primer apartado, uno de los servicios más completos y populares para la gestión y trabajo colaborativo con documentos *on line* es Google Docs. Vamos a verlo en profundidad, analizando sus características y posibilidades.

10.2.1 DOCUMENTOS DE TEXTO

- Crear o subir documentos ya creados de Word, Open Office, RTF, HTML o texto, así como guardarlos en nuestro ordenador con estos mismos formatos (además del formato PDF o ZIP).
- Aplicar formato a los documentos, utilizando para ello un sencillo editor del tipo WYSIWYG (*What You See Is What You Get*, es decir, lo que ves es lo que obtienes).
- Enviar por correo electrónico los documentos como archivos adjuntos.
- Trabajar de forma colaborativa y en tiempo real con otras personas, pudiendo invitarles mediante su dirección de correo electrónico para que puedan editar o solamente ver nuestros documentos y hojas de cálculo.
- Edición del historial de revisiones de nuestros documentos y hojas de cálculo, incluyendo la posibilidad de restaurar cualquier versión anterior.
- Publicar documentos *on line* para que estén disponibles para todo el mundo, como páginas web (mediante una URL creada de forma automática) o como documentos publicados en nuestro *blog*.

10.2.2 HOJAS DE CÁLCULO

- Importar y exportar datos con los formatos XLS, CSV y ODS (y exportar funciones para PDF y HTML).
- Aplicar formatos y fórmulas en las hojas de cálculo.
- *Chatear* en tiempo real con otros usuarios que estén editando nuestra hoja de cálculo.
- Insertar una hoja de cálculo, o una sección de una hoja de cálculo, en nuestro *blog* o página web.
- Crear de una forma muy sencilla gráficos de diversos tipos, a partir de los datos introducidos en las hojas de cálculo.

10.2.3 PRESENTACIONES

- Importar presentaciones que tengamos creadas en los formatos de archivo PPT y PPS.
- Guardar las presentaciones que realicemos en formato PPT o PDF.
- Editar y aplicar formato a las presentaciones, utilizando para ellos un sencillo editor del tipo WYSIWYG.
- Posibilidad de insertar imágenes y autoformas previamente creadas.
- Aplicar a las diapositivas diferentes tipos de temas para que se ajusten a nuestras preferencias.
- Compartir y editar presentaciones con otras personas (tanto con colaboradores con permisos de edición, como con lectores (sin permisos de edición)).
- Visualizar las presentaciones en tiempo real con otras personas, pudiendo interactuar con ellas mediante una herramienta de *chat*.
- Publicar las presentaciones en la Web.

10.3 CÓMO DARNOS DE ALTA EN EL SERVICIO GOOGLE DOCS

Si ya disponemos de una cuenta en Google (por ejemplo, si tenemos una dirección de correo electrónico en Gmail) no será necesario darnos de alta de nuevo para acceder al servicio de Google Docs, bastará con seleccionar el servicio, y nada más (pulsando para ello sobre el botón **Más** y seleccionando **Google Docs**).

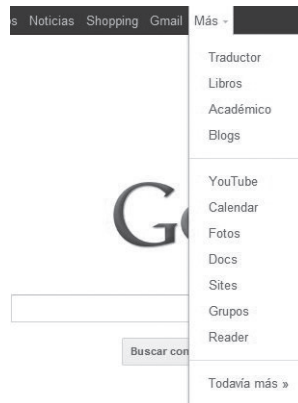


Figura 10.10. Google Docs

En la ventana que nos aparece, tendremos que introducir nuestra dirección de correo y, a continuación, la contraseña. Seleccionaremos **Recordarme en este equipo** si estamos utilizando un ordenador personal para no tener que introducir nuevamente la dirección de correo y la contraseña. Finalmente, tendríamos que pulsar sobre el botón **Acceder** (Figura 10.11).

 A screenshot of the Google login page. At the top right is a 'REGÍSTRSE' button. Below it is the 'Iniciar sesión' form. The form has fields for 'Correo electrónico' and 'Contraseña'. Below the password field is a checkbox labeled 'No cerrar sesión' which is checked. There is an 'Iniciar sesión' button. At the bottom, there is a link that says '¿No puedes acceder a tu cuenta?'.

Figura 10.11. Iniciar sesión

Si aún no tenemos cuenta Google en el Capítulo 4 se vio la forma de crear dicha cuenta.

10.4 CREACIÓN DE DOCUMENTOS

10.4.1 LA PANTALLA DE INICIO

Al iniciar Google Docs nos aparece la pantalla de inicio. Aquí podremos ver todos nuestros documentos:

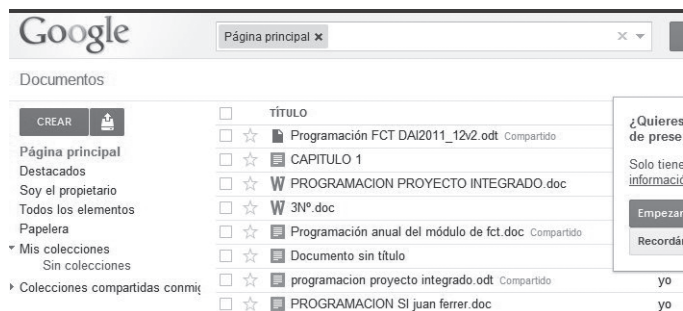


Figura 10.12. Google docs

10.4.2 NUEVO DOCUMENTO

Vamos a ver cómo crear un nuevo documento, por ejemplo un documento de texto. Para ello, haremos clic en el desplegable **Crear**. Google Docs nos permite la creación de seis elementos:

- Documento, un documento de texto.
- Presentación, basada en diapositivas.
- Hoja de cálculo.
- Formulario, con el que podemos recoger información de otros usuarios.
- Dibujo, donde podremos dibujar gráficos y diagramas.
- Colección, para organizar nuestros elementos.

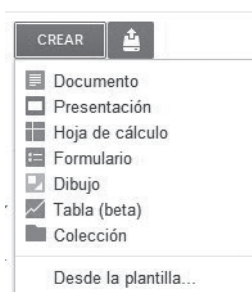


Figura 10.13. Nuevo

Como queremos crear un nuevo documento de texto, pulsamos en **Documento**. Se abrirá una nueva ventana o pestaña del navegador, en la que nos aparece una pantalla como ésta (en el caso de documentos de texto).

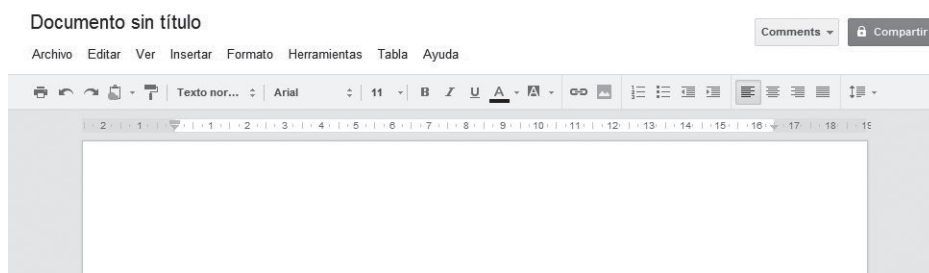


Figura 10.14. Nuevo

Dentro de la ventana del navegador, en la parte superior izquierda, nos aparece el título del documento (el nombre con el que se guardará), la fecha de su última modificación y el nombre del autor.

Al crear un documento nuevo, en el título aparece *Sin título*. La primera vez que se guarde, el título tomará el valor de la primera frase escrita. Para cambiarlo, haga clic sobre el título. Nos aparecerá una ventana pidiéndonos el nuevo nombre. Teclee “Primero”, que es el nombre que le vamos a dar a nuestro primer documento.

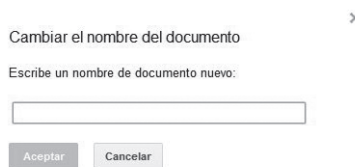


Figura 10.15. Nuevo

10.4.3 GUARDAR UN DOCUMENTO

Google Docs va guardando automáticamente su documento. Si ahora cerrásemos el navegador accidentalmente o apagásemos el ordenador, nuestro documento no se perdería y lo tendríamos disponible al iniciar una nueva sesión. De hecho va guardando automáticamente cada vez que añadimos o eliminamos algo de un documento.



Figura 10.16. Guardando



Figura 10.17. Cambios guardados

Por tanto, no deberemos preocuparnos de guardar el documento.

ACTIVIDADES 10.1



- Iniciar Google Docs, acceder a través de nuestra cuenta Google. Una vez iniciado Google Docs entrará en la pantalla de inicio. Haga clic sobre el botón **Crear nuevo** y pulse **Documento**; le aparecerá un documento de texto en blanco.
- Ahora copie el texto que ve a continuación:
"Al asomar la cabeza, quedó deslumbrado. Miró hacia abajo, hacia la penumbra de donde él surgía. Entre sus botas de goma negra, brillantes, vio el rostro de su compañero mal afeitado, prematuramente viejo.
Miró hacia abajo, hacia la penumbra de donde él surgía. Entre sus botas de goma negra, brillantes, vio el rostro de su compañero mal afeitado, prematuramente viejo".
- Guarde el documento como "apellido1_apellido2_nombre_practica1".

ACTIVIDADES 10.2



- Abra una nueva hoja de cálculo de Google Docs.
 - a. Sitúese en la celda A1 y escribe 1995. Pulse la tecla **Intro**.
 - b. Se encuentras en la celda A2. Escribe 1.995 y pulse la tecla **Intro**.
 - c. Sitúese en la celda A3 y escribe 1995,12. Pulse **Flecha Abajo**.
 - d. Escriba 1234567890123456789,1 en la celda A4. Pulse **Flecha Abajo**.
 - e. Escriba -2950 en la celda A5. Pulse **Intro**.
 - f. Sitúese en la celda A6 y escribe (2950). Pulse **Intro**.
 - g. Sitúese en la celda A7 y escribe 12%. Pulse **Intro**.
 - h. Sitúese en la celda A8 y escribe 12,7%. Pulse **Intro**.
 - i. Sitúese en la celda A9 y escribe 1200€. Pulse **Intro**.
 - j. Guarde el documento como "apellido1_apellido2_nombre_practica2".

10.5 COMPARTIR Y PUBLICAR

Sin duda, uno de los mayores atractivos que nos ofrece Google Docs es poder compartir nuestros documentos con otros usuarios. Podemos desde colaborar en su creación y edición con otros usuarios, hasta publicarlo con una dirección propia, como si de una página web se tratase.

Lo primero, y como nos referiremos a ellos con frecuencia, vamos a definir los tipos de participantes a la hora de compartir:

- **Propietario:** es el creador del documento. Puede editar el documento y eliminarlo, invitar a lectores y colaboradores, y cambiar algunos de sus derechos sobre el documento. Ningún colaborador puede eliminar la participación del propietario en el documento.

- **Colaboradores o Para editar:** son invitados por el propietario, aunque a su vez pueden invitar a otros colaboradores y lectores. Tienen derecho a leer, modificar, guardar e imprimir el documento.
- **Lectores o Para ver:** pueden leer el documento, guardárselo e imprimirlo, pero no editarlo.

10.5.1 COMPARTIR UN ELEMENTO

Podemos compartir cualquiera de nuestros elementos de Google Docs, incluso las carpetas. De hecho, al compartir una carpeta, se comparte todo su contenido. Por eso, si tenemos un grupo de personas con el que siempre compartimos documentos, nos será más cómodo compartir una carpeta e ir asignando los documentos a dicha carpeta, que ir compartiéndolos cada vez.

Los pasos a seguir para compartir un elemento son los siguientes:

1 Podemos compartir varios documentos a la vez desde el listado de la pantalla de inicio. Primero los seleccionamos y, después, pulsamos el botón **Compartir** (Figura 10.18).

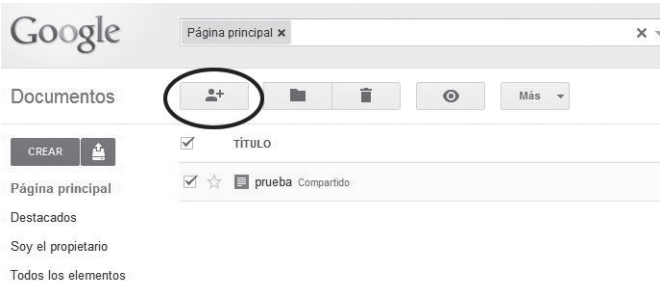


Figura 10.18. Compartir

2 Para compartir el documento que estamos editando, seleccionamos del botón **Compartir** (Figura 10.19), que mostrará las mismas opciones que para compartir desde la pantalla de inicio. En cualquier caso llegamos a la pantalla **Configuración para compartir** (Figura 10.20).

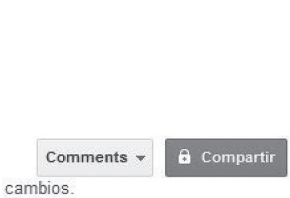


Figura 10.19. Compartir

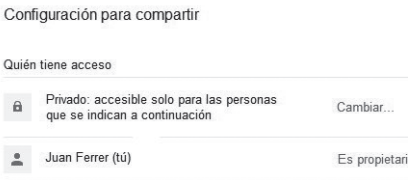


Figura 10.20. Configuración para compartir

3 Por defecto, el archivo es privado (Figura 10.20), esto implicará que solo será accesible por el propietario y por los usuarios con los que se ha compartido el documento de forma explícita, por nadie más. Podemos cambiar la seguridad de la compartición haciendo clic en **Cambiar** (Figura 10.21).

Configuración para compartir

Opciones de visibilidad:

- ☒ **Público en la Web**
Cualquier usuario de Internet puede encontrar el elemento y acceder a él; no es necesario iniciar sesión.
- ☐ **Cualquier usuario que reciba el enlace**
Cualquier usuario que reciba el enlace puede acceder al elemento; no es necesario iniciar sesión.
- ☐ **Privado**
Solo pueden acceder a los elementos las personas a quienes se les haya otorgado permisos de forma explícita. Se requiere iniciar sesión.

Nota: los elementos sin opción de visibilidad pueden seguir publicándose en la Web. [Más información](#)

[Más información acerca de la visibilidad](#)

Figura 10.21. Cambiar el acceso a un archivo

10.5.2 INVITAR A COLABORADORES Y LECTORES

Podemos invitar a nuevos colaboradores y lectores desde la pantalla **Configuración para compartir**, usando la pestaña **Añadir a personas** (Figura 10.23).

En la caja de texto, escribiremos las direcciones de correo de todos los usuarios, separadas por comas si hay más de una. Pulsando el enlace **Elegir de los contactos**, accedemos a una lista con nuestros contactos de Gmail, para poder añadir esas direcciones con un clic (Figura 10.22).

Lo normal es avisar al contacto que le hemos compartido con el un documento, es este caso marcaremos la opción **Notificar a las personas por correo electrónico**. Podemos personalizar este mensaje pulsando en el enlace **Añadir Mensaje**.

Por defecto, nuestros contactos puedan **editar** el documento compartido. Existen 2 tipos de compartición extra (puede comentar o puede ver). Dependiendo del nivel de participación que queramos mantener con la/s persona/s con la/s que compartimos el documento, utilizaremos una opción u otra (Figura 10.23).

Añadir a personas: Elegir de los contactos

Escribe nombres, direcciones de correo electrónico o grupos...

☒ Notificar a las personas por correo electrónico - Añadir mensaje

☐ Enviar una copia

☐ Pegar el elemento en el correo electrónico

...os editores pueden añadir a personas y cambiar los permisos. [\[Cambiar\]](#)

Figura 10.22. Compartir con los contactos

Añadir a personas: Elegir de los contactos

juanferer.profe@gmail.com

☒ Notificar a las personas por correo electrónico - Descartar mensaje

Opcional: incluir un mensaje personal...

☒ Puede editar

☐ Puede comentar

☐ Puede ver

☐ Enviar una copia

☐ Pegar el elemento en el correo electrónico

Figura 10.23. El contacto

Si marcamos **Pegar el elemento en el correo electrónico**, el documento se enviará en el cuerpo del mensaje, como si lo hubiésemos escrito dentro del *Mensaje*.

Enviarme una copia envía una copia a nuestro correo.

Pulsando **Compartir y guardar**, el documento ahora ha sido compartido con el/los contacto/s que hayamos intrucido en la caja texto. A su vez, los elementos compartidos aparecerán en el listado de documentos de nuestro colaborador.

Podemos compartir un documento con un máximo de 100 usuarios, pero solo 10 podrán editarlo de forma simultánea.

ACTIVIDADES 10.3



➤ Compartir con el profesor/a los documentos creados en las Actividades 10.1 y 10.2.

10.5.3 GESTIONAR EL USO COMPARTIDO

Partimos de **Configuración para compartir** (Figura 10.25), como hemos comentado antes. Por defecto es **Privado**, es decir, solo es accesible para las personas que se indican en la lista que aparece a continuación.

Podemos cambiar la **Configuración** pulsando sobre **Cambiar** (Figura 10.26).



Figura 10.24. Configuración para compartir

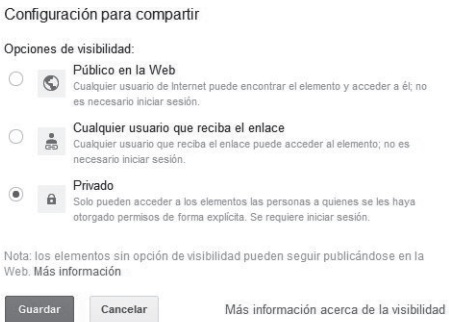


Figura 10.25. Opciones de visibilidad

En nuestro caso vamos a seleccionar **Cualquier usuario que reciba el enlace**, lo cual implicará que cualquier usuario, aunque no se haya identificado, que reciba el enlace podrá acceder a dicho elemento (Figura 10.27). Al igual que antes, podemos asignar ciertos permisos sobre los documentos compartidos (Modificar, Ver, Comentar). Generará un enlace que podremos distribuir para que los contactos puedan acceder a dicho contenido (Figura 10.28).

Configuración para compartir

Opciones de visibilidad:

- ☐ **Público en la Web**
Cualquier usuario de Internet puede encontrar el elemento y acceder a él; no es necesario iniciar sesión.
- ☒ **Cualquier usuario que reciba el enlace**
Cualquier usuario que reciba el enlace puede acceder al elemento; no es necesario iniciar sesión.
- ☐ **Privado**
Solo pueden acceder a los elementos las personas a quienes se les haya otorgado permisos de forma explícita. Se requiere iniciar sesión.

Acceso: Cualquier usuario (no requiere acceso) Puede ver ▼

Nota: los elementos sin opción de visibilidad pueden seguir publicándose en la Web. Más información

Guardar

Cancelar

Más información acerca de la visibilidad

Configuración para compartir

Enlace para compartir

<https://docs.google.com/document/d/1qFvb0fBdeDeOTGUof1xCKJozJC8uTh4XvtSQ2gPqp28/edit>

Comparte este enlace a través de: Google+ Gmail Facebook Twitter

Quién tiene acceso

	Cualquier usuario que reciba el enlace puede ver los elementos.	Cambiar...
	(tú) aplicacionesweb.ram...	Puede editar ▼ x
	juanferrer.profe...	Es propietario
		Puede editar ▼ x

Añadir a personas:

Escribe nombres, direcciones de correo electrónico o grupos...

Los editores pueden añadir a personas y cambiar los permisos.

Fin

Figura 10.26. Cualquier usuario que reciba el enlace

Figura 10.27. Enlace para compartir

Podremos obtener el enlace siempre que queramos accediendo a **Compartir**, desde **Documento** o desde la pantalla principal.

ACTIVIDADES 10.4



- Compartir con cualquiera que reciba el enlace los documentos creados en las Actividades 10.1 y 10.2. Enviar al profesor/a el enlace generado por Google tras compartir dichos elementos.

10.5.4 PÚBLICO EN LA WEB

Esta opción es parecida a la anterior, con la diferencia que en este caso cualquier usuario de Internet puede acceder al recurso compartido. No es necesario que inicie sesión, tan solo que conozca el enlace de dicho documento, de la misma manera podemos compartir el recurso de varias formas: Editar, Comentar y Ver.

Los pasos a seguir son los mismos que en el caso anterior, tan solo cambiando las opciones de visibilidad por **Público en la web**.

ACTIVIDADES 10.5



- Hacer públicos en la web los documentos creados en las Actividades 10.1 y 10.2. Enviar al profesor/a el enlace generado por Google tras compartir dichos elementos.

10.5.5 PUBLICAR UN DOCUMENTO

Publicar un documento en Google Docs es, como se dice coloquialmente, colgarlo en Internet. Para ello, dentro del documento, pulsamos en **Archivo, Publicar en la web** (Figura 10.28).

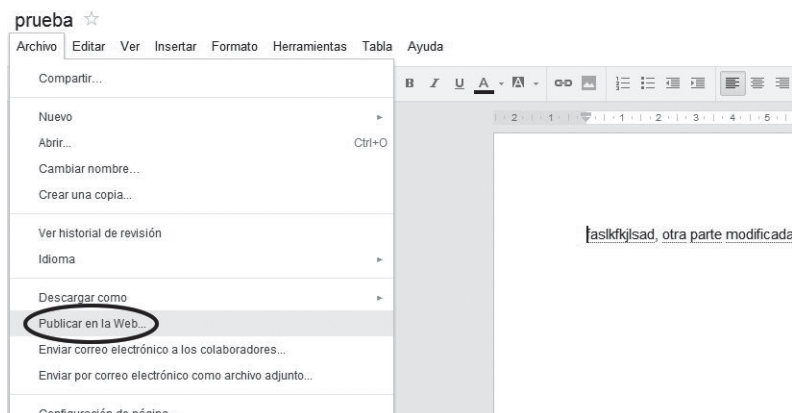


Figura 10.28. Archivo, Publicar en la web

La opción **Volver a publicar automáticamente cuando se realicen cambios**, sincroniza la versión publicada de nuestro documento con el documento que editamos. Así, los cambios realizados se irán actualizando en la versión publicada (solo para documentos de texto).



Figura 10.29. Opciones de Publicar en la web

Google Docs también nos permite publicar nuestro documento de texto a través de un *blog* o una web, mediante el enlace **Insertar código** podemos insertar este código en nuestro *blog* de la misma forma que insertamos música o presentaciones (ver Capítulo 4).

La principal diferencia entre publicar y el enlace para compartir, es que con el enlace muestra el documento actual, mientras que al publicar, publicamos una versión. Si no la actualizamos, será distinta al documento original.

En el caso de los documentos de texto, podemos consultar las estadísticas de visitas de sus documentos publicados si configuramos Google Analytics.

En las hojas de cálculo podemos decidir qué hoja publicar y en qué formato se visualizan (Figura 10.30). En las presentaciones, encontraremos un código HTML. Si lo copiamos en una página web se verá un marco con los controles necesarios para reproducir la presentación (Figura 10.31).

Publicar en la Web

Hojas para publicar

Todas las hojas ▾

☒ Volver a publicar automáticamente cuando se hagan cambios

Iniciar la publicación Volver a publicar ahora

Nota: la publicación de un documento no afecta a la opción de visibilidad. Más información

Obtener un enlace a los datos publicados

Página web ▾

Todas las hojas ▾

Todas las celdas

Iniciar la publicación para obtener el enlace

Cerrar

Figura 10.30. Publicar Hoja de cálculo

Publicar en la Web

Control de publicación

Detener publicación

Nota: el hecho de publicar un documento no afecta a la opción de visibilidad. Más información

Obtener un enlace al documento publicado

Enlace de documento

<https://docs.google.com/presentation/pub?id=19-pXg0lI8IzYQXOqoVMJHeo5m>

Insertar código

```
<iframe src="https://docs.google.com/presentation/embed?id=19-pXg0lI8IzYQXOqoVMJHeo5mGY_ejluW2BVfz7s4&start=false&loop=false&delays=3000" frameborder="0" width="960" height="749">
```

Si lo prefieres, comparte este enlace mediante: Google+ Gmail Facebook Twitter

Tamaño de la presentación

Mediana (960x749) ▾

Avanzar la presentación a la diapositiva siguiente de forma automática

cada 3 segundos (opción predeterminada) ▾

Figura 10.31. Publicar Presentación

ACTIVIDADES 10.6



- En este ejercicio deberá publicar en Internet la Actividad 10.2. Recuerde: aunque publique el documento, no será accesible por nadie que no conozca su dirección, por tanto, deberá enviar la dirección del documento al profesor/a.

10.6 DESCARGAR EL DOCUMENTO

Otra opción de la que disponemos es descargar el documento a nuestro disco duro, pudiéndolo editar después con otros programas.

Primero debemos elegir cómo lo guardaremos. Para ello debemos tener en cuenta qué vamos a hacer después con el documento y de qué programas disponemos para leerlo o modificarlo. Google Docs nos permite guardar nuestros documentos en los siguientes formatos:

- **Documentos de texto:** Word (.doc.), Open Office (.odt.), PDF (.pdf), RTF (.rtf), Texto sin formato (.txt), HTML comprimido (.zip).
- **Presentaciones:** PDF (.pdf), PowerPoint (.ppt), Texto (.txt).
- **Hojas de cálculo:** Excel (.xls), OpenOffice (.ods), PDF (.pdf), CSV (.csv), Texto (.txt), Html (.html).
- **Dibujos:** PNG, JPEG, SVG, PDF.
- **Documentos PDF:** PDF (.pdf).

ACTIVIDADES 10.7



- Guarde las Actividades 10.1 y 10.2 como documentos de Microsoft Word y Excel, respectivamente. Envíe dichos documentos a su profesor/a.

ACTIVIDADES 10.8



- Guarde las Actividades 10.1 y 10.2 en formato PDF. Envíe dichos documentos a su profesor/a.



RESUMEN DEL CAPÍTULO

En este capítulo hemos tratado el futuro hecho presente, las aplicaciones de ofimática web, que tienen poco que envidiar a sus hermanas mayores, las aplicaciones ofimáticas tradicionales y, además, presentan una serie de ventajas sobre ellas. En primer lugar, la mayoría son gratuitas, aunque no es el factor diferenciador, ya que existen aplicaciones de ofimática de escritorio también gratuitas. En segundo lugar, los documentos están alojados en la nube, por tanto, para acceder a ellos no es necesario llevar ninguna memoria *flash* u ordenador. Tan solo necesitamos un navegador y una conexión a Internet. Este hecho determina que puede haber varios usuarios a la vez trabajando sobre el mismo documento, cosa impensable en las aplicaciones ofimáticas de escritorio. En tercer lugar, debido a que solo necesitamos un navegador y una conexión a Internet, podremos utilizarlo desde cualquier dispositivo que cumpla estos requisitos: ordenadores, *smartphones*, *tablets*, *netbooks*, televisiones, etc.

No todo iban a ser alegrías: debido a la necesidad de una conexión a Internet, si perdemos la conexión, perderemos también el acceso a los documentos que hemos creado, aunque todas las aplicaciones llevan la opción de guardar una copia del documento en local. A su vez, al funcionar desde el navegador tampoco pueden implementar por completo las funcionalidades de las aplicaciones ofimáticas de escritorio.



EJERCICIOS PROPUESTOS

1. ¿Qué diferencias principales encuentras entre las aplicaciones ofimáticas de escritorio (Office, Libre Office) y las aplicaciones ofimáticas web?
2. Vista la tendencia actual con los *smartphones*, *tablets*, *netbooks*, etc., ¿cree que el futuro de la ofimática serán las aplicaciones ofimáticas web?
3. Regístrese y pruebe tanto Zoho como ThinkFree, ¿qué le parecen? ¿Cree que alguna de ellas desbancará a la todopoderosa Google? ¿Por qué?
4. Una de las ventajas más grandes de las aplicaciones ofimáticas web es que varios usuarios pueden editar a la vez un documento. ¿Qué ventajas e inconvenientes encuentra en esto?
5. Localice en Internet alguna aplicación de ofimática web, aparte de las que hemos visto en el capítulo.



TEST DE CONOCIMIENTOS

1 La aplicación ofimática web más utilizada hoy en día es:

- a) Zoho.
- b) Microsoft Office.
- c) Libre Office.
- d) ThinkFree.
- e) Gmail.
- f) Todas son ciertas.
- g) Todas son falsas.

2 La aplicación ofimática web en la que necesitamos tener la máquina virtual de Java en nuestro sistema para poder utilizarla es:

- a) Zoho.
- b) Microsoft Office.
- c) Libre Office.
- d) ThinkFree.
- e) Gmail.
- f) Todas son ciertas.
- g) Todas son falsas.

3 Ventajas de las aplicaciones de ofimática web con respecto a las aplicaciones de ofimática de escritorio:

- a) Multiplataforma.
- b) Varios usuarios a la misma vez en un mismo documento.
- c) No necesitamos gastar espacio de almacenamiento en nuestro dispositivo para guardar los documentos.
- d) Tenemos siempre la última versión del programa.
- e) Todas son falsas.

4 Ventajas de las aplicaciones ofimáticas de escritorio frente a las aplicaciones ofimáticas web:

- a) Mucho más potentes y con más funcionalidades.
- b) Podemos utilizarlas sin conexión a Internet.
- c) Podemos hacer archivos. mucho más grandes.
- d) Todas son ciertas.
- e) Todas son falsas.

5 Con Google Docs podemos exportar documentos que hemos creado a formato:

- a) PDF.
- b) Office.
- c) Open Document.
- d) Todas son ciertas.
- e) Todas son falsas.

11

Diseño del contenido y la apariencia de documentos web

OBJETIVOS DEL CAPÍTULO

- ✓ Conocer el lenguaje de programación web de marcas por excelencia: HTML.
- ✓ Conocer el lenguaje de hojas de estilo en cascada CSS.
- ✓ Practicar sobre ambos lenguajes.

11.1 LENGUAJES DE MARCAS PARA REPRESENTAR EL CONTENIDO DE UN DOCUMENTO

11.1.1 ¿HTML?

HTML, siglas de *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>).

11.1.2 HERRAMIENTAS PARA EL DESARROLLO WEB

Cuando se van a desarrollar páginas web básicas, no es necesario tener gran cantidad de recursos: basta con un editor de texto y un navegador web.

Tipos de editores

Se puede establecer una división en dos grupos de los tipos de aplicaciones que permiten desarrollar páginas web: los editores de texto y las aplicaciones WYSIWYG (*What You See Is What You Get* o Lo que ves, es lo que obtienes).

Los editores de texto son programas que permiten escribir y modificar archivos digitales compuestos por texto plano (si pueden también manejar texto con formato, tablas, imágenes y otros elementos complejos, son denominados procesadores de texto). Los editores de texto pueden ser, a su vez, simples o dedicados a la creación de páginas web o a la programación en general.

En el caso de los editores simples, los mejores para el desarrollo de páginas web en cualquiera de los lenguajes posibles (HTML, PHP, etc.) son aquellos que precisamente menos funcionalidades ofrecen a la hora de editar los textos.

En cuanto a los editores de texto destinados a la programación, suelen incorporar algunas funciones muy útiles para el desarrollador. Algunos pueden resaltar palabras reservadas o especiales del lenguaje de programación, proporcionar listados de etiquetas, introducir instrucciones de forma automática, etc.

Las aplicaciones WYSIWYG, por su parte, son programas especializados en los que se trabaja de forma gráfica. El desarrollador trabaja directamente con los elementos de la página web (*banners*, menús, etc.), siendo la aplicación la encargada de componer la estructura de código correspondiente en el lenguaje elegido.

Características

Facilidad de aprendizaje. El aprendizaje más sencillo y rápido es el de las aplicaciones WYSIWYG, ya que se trabaja de forma gráfica y no hay que conocer profundamente el lenguaje que se utiliza. En los editores simples y especializados para la programación es necesario conocer tanto la estructura de un documento elaborado en dicho lenguaje, como los elementos del mismo y la forma en que se usan.

11.1.3 NUESTRA PRIMERA PÁGINA EN HTML

A continuación se presenta el código fuente de nuestra primera página web:

PRÁCTICA 11.1

```
<HTML>
<HEAD>
<TITLE>
Mi primera pagina
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>
Mi Primera pagina
</H1>
</CENTER>
<P>
Bienvenido a mi primera pagina Web
</BODY>
</HTML>
```

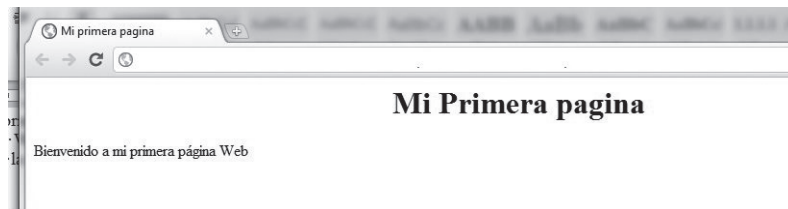


Figura 11.1

Varias cosas a comentar sobre la Práctica 11.1. Lo primero que conviene explicar es en qué consisten todos esos símbolos de mayor y menor que están distribuidos por ahí. El lenguaje HTML se basa en la sintaxis SGML. Esto quiere decir que cualquier cosa que hagamos en HTML estará encerrada entre dos etiquetas de esta manera:

```
<ETIQUETA parámetros>
...
</ETIQUETA>
```

Toda página HTML que se precie serlo, debe incluir obligatoriamente la etiquetas `<HTML>` y `</HTML>`, estas etiquetas nos están indicando que el código contenido entre ellas va a ser HTML.

Un documento HTML tiene una estructura que lo separa en dos partes: cuerpo y cabecera. Toda página HTML debe de incluir las etiquetas `<HEAD>` `</HEAD>`, que representan la cabecera de la página, y `<BODY>` `</BODY>`, que representan el cuerpo de la página. Dentro de BODY estará la página en sí, mientras que en HEAD incluiremos algunas cosas que no se ven al principio pero que pueden llegar a ser muy importantes. En primer lugar irá definida la cabecera y a continuación el cuerpo.

Vemos el código de HEAD:

```
<HEAD>
<TITLE>
Mi primera pagina
</TITLE>
</HEAD>
```

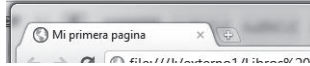


Figura 11.2

Dentro de la cabecera solo hay otra etiqueta: el título de la página. Es lo que veremos como título de la ventana en los navegadores que lo permitan. Es como se conocerá nuestra página en algunos buscadores y en la agenda de direcciones (*bookmarks*) de los usuarios. Por lo tanto, parece importante pensarnos bien cómo llamarla. Este título aparecerá en la barra superior del navegador junto con el nombre de dicho navegador.

Ahora vamos a indicar el contenido. Lo primero será indicar que estamos en el cuerpo del documento:

```
<BODY>
...
</BODY>
```

Con la etiquetas `<CENTER>` y `</CENTER>` indicamos que el texto que está comprendido entre estas etiquetas irá centrado en la pantalla de navegador.

Las etiquetas `<H1>` y `</H1>` indican el tamaño del texto, en este caso *cabecera1*, el más grande de todos, lo cual nos permitirá verlo más grande que el resto de la página web.

Por último, la etiqueta `<P>` nos indica que el texto que viene a continuación es un párrafo normal y corriente, el típico texto que nos vamos a encontrar en cualquier página web, en contraposición a `<H1>` como se podrá apreciar cuando se pruebe esta Práctica 11.1. Cabe destacar que la etiqueta `<P>` no tiene contraposición `</P>` en nuestra práctica, podría tenerlo una vez acabado el párrafo al que afecta, aunque no es necesario y, por tanto, no se incluye.

Para probar la práctica podemos utilizar el *Bloc de notas* de Windows o el editor de textos *gedit* de Ubuntu. La guardamos como *Caso práctico1.html* y ya podemos probarla haciendo doble clic sobre él.

ACTIVIDADES 11.1



➤ Crear nuestra primera página web con las siguientes consideraciones:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto del encabezado1 (`<h1>`) debe ser *primera práctica de nombre y apellidos del alumno/a*.
- El texto del cuerpo de la página debe mostrar: Esta es mi primera página web realizada para implantación de aplicaciones web.

11.1.4 ETIQUETAS HTML BÁSICAS

- **Etiquetas de formato de texto:** vamos a ver las principales etiquetas, que nos permite dar formato a un texto (o a una parte de él):
 - **Negrita:** para poner una parte (o la totalidad) del texto en negrita se utilizan las etiquetas y .
 - **Cursiva:** para poner una parte (o la totalidad) del texto en cursiva (también se le conoce como Itálica) se utilizan las etiquetas <I> y </I>.
 - **Teletipo:** para poner una parte (o la totalidad) del texto en teletipo se utilizan las etiquetas <TT> y </TT>.
 - **Tachado:** para que una parte (o la totalidad) del texto aparezca tachado se utilizan las etiquetas <S> y </S>.
 - **Subrayado:** para que una parte (o la totalidad) del texto aparezca subrayado se utilizan las etiquetas <U> y </U>.
 - **Dirección de correo electrónico:** para poner una parte (o la totalidad) del texto con formato de dirección de correo electrónico se utilizan las etiquetas <ADDRESS> y </ADDRESS>.
 - **Sangría a un párrafo:** para poner que el texto aparezca con sangría debemos utilizar las etiquetas <BLOCKQUOTE> y </BLOCKQUOTE>.

PRÁCTICA 11.2



- Uso de negrita y cursiva, por separado y juntas.

```
<HTML>
<HEAD>
<TITLE>
Segunda pagina Web
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>
Segunda pagina Web
</H1>
</CENTER>
<P> Aquí se puede <B>apreciar</B> la diferencia existente <I>entre el texto</I> en
<B>negrita</B> y el texto en <I>cursiva</I>, también puede haber una <B><I>mezcla
entre ambos</I></B>
</BODY>
</HTML>
```

Hay que tener en cuenta que en el caso de que haya varias etiquetas que afectan a una parte del texto (o toda) las etiquetas que determinan el fin de dicho formato, `</etiqueta>`, deben aparecer en orden inverso a como aparecieron afectando al formato. En nuestro caso aparecen `<B><I>`, dos etiquetas de formato afectando a “mezcla entre ambos”. Por lo tanto, el orden de fin de etiquetas debe ser inverso a como aparecieron, es decir, primero será `</I>` la que aparezca y después `</B>`.



Figura 11.3

ACTIVIDADES 11.2



➤ Nuestra segunda tarea debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto del encabezado1 (`<h1>`) debe ser *segunda práctica de nombre y apellidos del alumno/a*.
- El texto del cuerpo de la página debe contener: un párrafo (de 2 ó 3 líneas) en cursiva, otro párrafo en negrita y un último párrafo en negrita y cursiva.

PRÁCTICA 11.3



➤ Uso de subrayado, tachado, teletipo, dirección de correo electrónico, negrita y cursiva:

```
<HTML>
<HEAD>
<TITLE>
Tercera pagina Web
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>
Tercera pagina Web
</H1>
</CENTER>
<P>
<I>Con diez cañones por banda,</I>
<BR>
<S>viento en popa</S>,
<BR>
```

```

<U>a toda vela</U>,
<BR>
<TT>no corta el mar, </TT>
<BR>
<B>sino vuela,</B>
<BR>
<I>un velero bergantín.</I>
<BR>
<P>
<BLOCKQUOTE>
<U><B>Bajel pirata que llaman, </B></U>
<BR>
<S><I>por su bravura,</I></S>
<BR>
<TT><U>El Temido,</U></TT>
<BR>
<B>en todo mar conocido,</B>
<B><I>del uno al otro confín.</I></B>
</BLOCKQUOTE>
<BR>
<BR>
<P>Para contactar conmigo manda un email a:
<ADDRESS>prueba@prueba.com</ADDRESS>
</BODY>
</HTML>

```

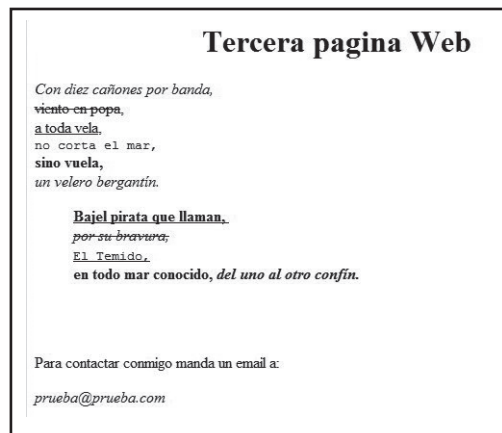


Figura 11.4

Aparece una nueva etiqueta `<BR>` no vista anteriormente, esta etiqueta va a producir un salto de línea (Intro) en nuestra página web. Como se ve en la práctica, no le corresponde ningún `</BR>` ya que esta etiqueta no afecta a ningún párrafo concreto, puede aparecer varias veces seguidas, produciendo tantos saltos de línea como veces aparezca.

En este caso abrimos y cerramos las etiquetas en la misma línea, por ejemplo: `<I>un velero bergantín.</I>`, esto es común cuando el párrafo afectado es corto.

ACTIVIDADES 11.3



» Nuestra tercera tarea debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto del encabezado1 (<h1>) debe ser *práctica tres de nombre y apellidos del alumno/a*.
- El texto del cuerpo de la página debe contener: un párrafo (de 2 ó 3 líneas) con el texto tachado, otro párrafo con el texto subrayado y otro párrafo con sangría y un último párrafo con el texto con formato de teletipo.

ACTIVIDADES 11.4



» Nuestra cuarta tarea debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto del encabezado1 (<h1>) debe ser *práctica cuatro de nombre y apellidos del alumno/a*.
- El texto del cuerpo de la página debe contener: un párrafo (de 2 ó 3 líneas) con el texto en negrita y tachado, otro párrafo con él en cursiva y subrayado y un último párrafo con el texto con formato teletipo y con sangría.

Etiquetas de encabezados

Ya vimos en la primera Práctica (11.1) una etiqueta de encabezado <H1>, en este apartado las vamos a ver todas:

- <H1> y </H1>: es el encabezado más grande de todos
- <H2> y </H2>: menor que <H1>
- <H3> y </H3>: menor que <H2>
- <H4> y </H4>: menor que <H3>
- <H5> y </H5>: menor que <H4>
- <H6> y </H6>: menor que <H5>

PRÁCTICA 11.4



» A continuación, en la siguiente práctica se muestra el resultado de aplicar varias etiquetas de encabezados:

```
<HTML>
<HEAD>
<TITLE>
Cuarta pagina Web
</TITLE>
</HEAD>
<BODY>
<H1> Cuarta pagina Web (Encabezado H1) </H1>
```

```

<BR>
<H2>Encabezado H2</H2>
<H3>Encabezado H3</H3>
<H4>Encabezado H4</H4>
<H5>Encabezado H5</H5>
<H6>Encabezado H6</H6>

<BR>

<P>Este es una parrafo normal para comparar con los encabezados

</BODY>

</HTML>

```

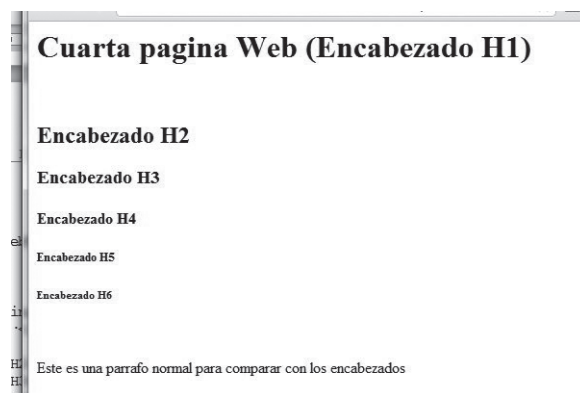


Figura 11.5

Como hemos apuntado anteriormente, podemos abrir y cerrar etiquetas de formato en la misma línea. Aparece de nuevo `<BR>` (un Intro), en este caso para separar de manera diferenciada el encabezado H1 del resto de encabezados. Nótese que cuando cambio de un estilo a otro se produce un salto de línea de manera automática, sin tener que especificar `<BR>`.

ACTIVIDADES 11.5



➤ Nuestra quinta tarea debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto del encabezado1 (`<h1>`) debe ser *práctica cinco de nombre y apellidos del alumno/a, encabezado h1*.
- El texto del encabezado2 (`<h2>`) debe ser *práctica cinco de nombre y apellidos del alumno/a, encabezado h2*.
- El texto del encabezado3 (`<h3>`) debe ser *práctica cinco de nombre y apellidos del alumno/a, encabezado h3*.
- El texto del encabezado4 (`<h4>`) debe ser *práctica cinco de nombre y apellidos del alumno/a, encabezado h4*.
- El texto del encabezado5 (`<h5>`) debe ser *práctica cinco de nombre y apellidos del alumno/a, encabezado h5*.
- El texto del encabezado6 (`<h6>`) debe ser *práctica cinco de nombre y apellidos del alumno/a, encabezado h6*.

PRÁCTICA 11.5



➤ Aquí tenemos otra práctica de encabezados, mezclados con párrafos:

```
<HTML>
<HEAD>
<TITLE>Quinta pagina
Web</TITLE>
</HEAD>
<BODY>
<H2> Encabezado </H2>
<P>Primer párrafo
debajo del encabezado.
<H3> Un encabezado <BR> en dos
renglones </H3>
<P>Párrafo con partes
<B>en negrita</B>
</BODY>
</HTML>
```

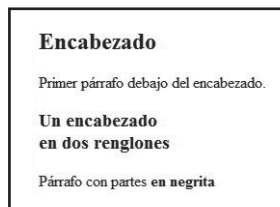


Figura 11.6

Acentos: *&_acute;* donde el espacio representa la vocal que va a aparecer tildada. En nuestro caso queremos que aparezca Párrafo en nuestra página web, deberemos escribir *P´rrafo*. Otro ejemplo: para que aparezca la palabra camión correctamente tildada (como se presenta aquí), deberemos escribir *camión* en nuestra página web.

Podemos observar un Intro dentro de un encabezado H3, a pesar de ese Intro no pierde el formato y además de que hay un párrafo en negrita.

ACTIVIDADES 11.6



➤ Nuestra sexta tarea debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica seis de nombre y apellidos del alumno/a*, encabezado <h1>.
- Un párrafo (2 ó 3 líneas), con texto normal (<p>).
- El texto con encabezado2 (<h2>) debe ser *práctica seis de nombre y apellidos del alumno/a*, (Intro) (Intro) encabezado <h2>.
- Un párrafo (2 ó 3 líneas), con texto normal (<p>).

ACTIVIDADES 11.7



➤ Nuestra séptima tarea debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica siete de nombre y apellidos del alumno/a*.
- Un párrafo que contenga las siguientes palabras: línea, además, página, camión, continuación, inserción, gráfico.

11.1.5 ETIQUETAS HTML AVANZADAS

Inserción de líneas de separación: si queremos introducir líneas para separar unos contenidos de otros debemos utilizar <HR>, el funcionamiento de <HR> es similar al de
 (Intro) y funciona de manera parecida, ya que
 introduce un salto de línea y <HR> lo que hace es dar un salto de línea pero, además, incluyendo una línea separatoria. Al igual que en el caso de
, tampoco existe </HR>.

PRÁCTICA 11.6



➤ En la práctica siguiente se pueden apreciar claramente las diferencias existentes entre
 y <HR>.

```
<HTML>
<HEAD>
<TITLE>Sexta página Web </TITLE>
</HEAD>
<BODY>
<H2> Encabezado 1 </H2>
<P>Primer párrafo debajo del encabezado.
<HR>
<H2> Encabezado 2 </H2>
<P>Segundo párrafo debajo del encabezado.
<BR>
<P> aquí seguiría el resto de la página
</BODY>
</HTML>
```

Encabezado 1

Primer párrafo debajo del encabezado.

Encabezado 2

Segundo párrafo debajo del encabezado.

aquí seguiría el resto de la página

Figura 11.7

ACTIVIDADES 11.8



» Nuestra octava tarea debe tener:

- a. El título de la página debe ser el nombre y apellidos del alumno/a.
- b. El texto con encabezado1 (<h1>) debe ser *práctica ocho de nombre y apellidos del alumno/a*.
- c. Una línea de separación.
- d. Un párrafo de 2 ó 3 líneas con texto normal.
- e. Una línea de separación.
- f. Un párrafo de 2 ó líneas con texto en negrita.
- g. Una línea de separación.
- h. Un texto con encabezado3 (h3) donde debe poner *fin de la práctica 8*.

Etiquetas y : las etiquetas y , son etiquetas de formato de texto y bien podrían haber sido estudiadas en el apartado 11.1.4, pero se ha considerado estudiarlas aparte debido a su importancia y a la cantidad de opciones aplicables a esta etiqueta (de ahí el hecho de comentarla en solitario).

La etiqueta se emplea para formatear el tipo y tamaño del texto en algunos puntos del documento específicos (como ya hemos indicado). En realidad, si la etiqueta abriera y cerrara el documento, todo el texto comprendido dentro quedaría formateado tal y como estuviera indicado. Sin embargo, conceptualmente, la etiqueta ha sido concebida para definir partes limitadas de texto.

Los atributos que puede tener se ven reflejados en la siguiente tabla:

Tabla 11.1 Atributos de FONT

Atributo	Propósito	Ejemplo
size="numero"	Define el tamaño de la fuente	size="2"
size="+numero"	Aumenta el tamaño de la fuente lo que indique el número	size="+2"
size="-numero"	Disminuye el tamaño de la fuente lo que indique el número	size="-2"
face="nombre fuente"	Establece la fuente	face="Times"
color="valor de color"	Define el color a través de su valor hexadecimal	color="#eeff00"
color="nombre de color"	Define el color a través de su nombre	color="red"

PRÁCTICA 11.7



» Vamos a ver la Práctica 11.7 sobre la utilización de la etiqueta :

```
<HTML>
<HEAD>
<TITLE>
```

```
p&aacute;gina Web 7
</TITLE>
</HEAD>
<BODY>
<H1> Uso de la etiqueta <font> </H1>
<BR>
<font size="4" color="red" face="Arial"> Prueba1 de
<font> </font>
<BR>
<font size="8" color="blue" face="Verdana"> Prueba2 de
<font> </font>
<HR>
<font size="5" color="green" face="Times New Roman">
Pruebal de <font> </font>
</BODY>
</HTML>
```



Figura 11.8

Tabla 11.2 Colores en HTML

Color	Código hexadecimal	Nombre del color
Rojo	0xFF0000	Red
Verde	0x00FF00	Green
Azul	0x0000FF	Blue
Blanco	0xFFFFFFFF	White
Cyan	0x00FFFF	Cyan
Magenta	0xFF00FF	Magenta
Amarillo	0xFFFF00	Yellow
Negro	0x000000	Black
Gris	0x808080	Gray
Rosa	0xFF8080	Pink
Violeta	0x800080	Violet
Marron	0x800000	Maroon
Naranja	0xFF8000	Orange
Turquesa	0x00FF80	Turquoise
Azul marino	0x000080	Navy

El uso de con los atributos vistos en la Figura 11.8: aparecen dos nuevos valores a colación: "<" y ">". Estos valores representan < (*less than*) y > (*greater than*), respectivamente. Hay que tener en cuenta que si quisiéramos por algún motivo escribir dentro de una página web la palabra , en lugar de aparecer esto lo que haría sería poner el texto a continuación en negrita. Para conseguir que muestre debemos utilizar . El atributo color puede contener el código hexadecimal o el nombre del color (en inglés). La tabla 11.2 muestra los códigos y los nombres de los colores más usuales:

La etiqueta NO debería ser usada. El Consorcio World Wide Web (W3C) ha eliminado la etiqueta en sus recomendaciones. En versiones futuras de HTML, las hojas de estilo (CSS) serán las utilizadas para definir las propiedades de los elementos HTML.

ACTIVIDADES 11.9



- Nuestra novena tarea debe tener:
- a. El título de la página debe ser el nombre y apellidos del alumno/a.
 - b. El texto con encabezado1 (<h1>) debe ser *práctica nueve de nombre y apellidos del alumno/a*, el color de este encabezado deberá de ser azul marino.
 - c. Una línea de separación.
 - d. La práctica debe contener un párrafo de 3 ó 4 líneas con texto intercalando variedades de FONT, mezclando tamaños de letra con colores.

Caracteres especiales: hasta este momento ya hemos visto algunas prácticas sobre caracteres especiales, como *&_acute*, *&lt* y *&gt*. En este apartado vamos a ver los principales caracteres especiales de HTML.

Tabla 11.3 Caracteres especiales

Código	Resultado
¿	¿
¡	¡
º	º
ª	ª
™	™
©	©
®	®
 	Espacio en blanco
<	<
>	>
&	&
"	"
&_acute;	Acento

Como práctica valdría cualquiera de las realizadas anteriormente que utilice estos caracteres especiales.

ACTIVIDADES 11.10

➤ Nuestra décima tarea debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica diez de nombre y apellidos del alumno/a*.
- La práctica también debe tener un párrafo con el siguiente contenido: "¿Quién ha osado a violar el © de mi marca ® iesto es un ultraje! " <preguntó él> "Parece que JJ & AA ha violado la patente ™ de nuestra marca" <contestó el abogado>.

11.1.6 ENLACES, VÍNCULOS O LINKS

Un enlace es una zona de texto o gráficos que si son seleccionados nos trasladan a otra página o a otra posición dentro de la página actual. Para incorporar un enlace hay que utilizar la etiqueta <A>. Todo lo que encerremos entre <A> y , ya sea texto o imágenes, será considerado como enlace y sufrirá dos modificaciones:

Se visualizará de manera distinta en el navegador. El texto aparecerá subrayado y de un color distinto al habitual, y las imágenes estarán rodeadas por un borde del mismo color que el del texto del enlace. Al pulsar sobre el enlace, seremos enviados a la página que apuntaba el enlace. Para que el enlace sirva para algo debemos especificarle una dirección. Lo haremos de la siguiente manera:

```
<A HREF="dirección">P&uacute;lsame </A>
```

El campo dirección puede ser de dos tipos, otra página web o una referencia a la misma página en la que estamos. Cuando la referencia es a otra página web, debemos indicar la dirección y página a la que hace referencia. En este caso tenemos a su vez dos posibilidades: que la página este dentro de mi sitio web o que sea externa. En el primer caso podré utilizar la ruta relativa y en el segundo tendré que utilizar la ruta absoluta obligatoriamente.

Cuando la referencia es dentro de mi misma página debo utilizar las conocidas como anclas, en este caso la forma del campo dirección será así: #nombre_ancla. A su vez, dentro de la página web debe existir una etiqueta con el nombre del ancla de la forma siguiente: .

PRÁCTICA 11.8

➤ Esta práctica enlaza 2 páginas, una externa a la mía y la otra dentro de mi sitio web.

```
<HTML>
<HEAD>
<TITLE>
p&aacute;gina Web8
</TITLE>
</HEAD>
<BODY>
<H1> Links
</H1>
<BR>
<A HREF="http://www.google.es">pincha aqu&iacute; para ir a
```

```
google</A>
<BR>
<BR>
<A HREF="ejemplo7.html">pincha aquí para ir al Caso
práctico7</A>
</BODY>
</HTML>
```

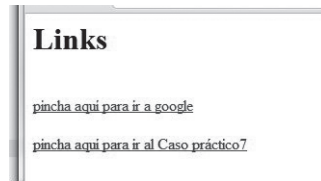


Figura 11.9

Cuando se haga clic en el primer enlace nos abrirá una página externa a nuestro sitio web, mientras que en el segundo enlace nos abrirá una página que pertenece a nuestro sitio web.

ACTIVIDADES 11.11



➤ Nuestra tarea número 11 debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica once de nombre y apellidos del alumno/a*.
- La página web de la práctica debe contener enlaces a las siguientes webs (la primera columna presenta la web a la que se referencia, la segunda el texto que debe aparecer en el enlace):

- http://www.google.es	→	Buscador google
- http://www.bing.es	→	Buscador bing
- http://www.ask.es	→	Buscador ask
- http://www.mozilla-europe.org	→	Página oficial de mozilla
- http://www.gmail.com	→	Página oficial de Gmail

ACTIVIDADES 11.12



➤ Nuestra tarea número 12 debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica doce de nombre y apellidos del alumno/a*.
- La página web de la práctica debe contener enlaces a las prácticas que hemos realizado hasta el momento, de la forma "Pulse aquí para ir a la Actividad 11.1", de forma que al hacer clic nos abra nuestra primera tarea realizada.

ACTIVIDADES 11.13



- Nuestra tarea número 13 debe tener:
- a. El título de la página debe ser el nombre y apellidos del alumno/a.
 - b. El texto con encabezado1 (<h1>) debe ser *práctica trece de nombre y apellidos del alumno/a*.
 - c. La página debe contener una etiqueta al principio y al final identificando cada parte como tal, hacer una página lo suficientemente larga y colocar un enlace al principio de la web para ir al final y otro al final de la web para ir al principio.

11.1.7 IMÁGENES

Podemos insertar imágenes en una página web, que nos ayudarán a mejorar el aspecto visual de una página web. Para insertar una imagen es necesario insertar la etiqueta . Dicha etiqueta no necesita etiqueta de cierre, como sucede con
 y <HR>. El nombre de la imagen ha de especificarse a través del atributo “src”. Por lo tanto, para insertar una imagen en una página web habrá que escribir:

```
<IMG SRC="imagen.gif">
```

Para trabajar de una forma más sencilla y ordenada, es recomendable que todas las imágenes se encuentren en un mismo directorio. En la siguiente tabla se indican los principales atributos de una imagen y su propósito:

Tabla 11.4 Propiedades de img en HTML

Etiqueta	Propósito	Ejemplo
border	Puede tomar valores numérico, que indican el grosor en pixeles del borde.	border="4"
width	Ancho de una imagen. El valor ha de ser un número bien pixeles, bien acompañado de % cuando se desee que se en un porcentaje con respecto a la página.	width="200"
height	Alto de una imagen. El valor ha de ser un número bien pixeles, bien acompañado de % cuando se desee que se en un porcentaje con respecto a la página.	height="80"
align	Alineacion de las imágenes con respecto a la línea de texto en la que se encuentran. Puede ser <i>left</i> (izquierda), <i>right</i> (derecha), <i>top</i> (alto), <i>bottom</i> (bajo), <i>middle</i> (centrado).	align="middle"

PRÁCTICA 11.9



```
<HTML>
<HEAD>
<TITLE> Web
</TITLE>
</HEAD>
<BODY>
<H1> Web</H1>
<BR>
<BR>
<BR>
<IMG SRC="IMG1.GIF" WIDTH="280" HEIGHT="210" BORDER="3">
<BR>
<BR>
<IMG SRC="IMG1.GIF" WIDTH="680" HEIGHT="210" BORDER="0">
</BODY>
</HTML>
```



Figura 11.10

ACTIVIDADES 11.14



» Nuestra tarea número 14 debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica catorce de nombre y apellidos del alumno/a*.
- Buscar en la Web al menos 3 imágenes e insertarlas en nuestra actividad utilizando los diversos atributos que se han visto para las imágenes.

Una variación de esto, sería poner una imagen como fondo de nuestra página web. En este caso debemos utilizar la propiedad BACKGROUND de la etiqueta <BODY>. Por tanto, la cosa quedaría tal que así:

```
<BODY BACKGROUND="FONDO.GIF">
```

Modificando la práctica anterior, añadiendo el atributo BACKGROUND a <BODY>, nos quedaría una práctica como la que sigue:

PRÁCTICA 11.10



```
<HTML>
<HEAD>
<TITLE> Web </TITLE>
</HEAD>
<BODY BACKGROUND="FONDO.GIF">
<H1> Undeacute;cima p&aacute;gina Web</H1>
<BR>
<BR>
<IMG SRC="IMG1.GIF" WIDTH="280" HEIGHT="210" BORDER="3">
<BR>
<BR>
<IMG SRC="IMG1.GIF" WIDTH="680" HEIGHT="210" BORDER="0">
</BODY>
</HTML>
```

ACTIVIDADES 11.15



➤ Nuestra tarea número 15 debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica quince de nombre y apellidos del alumno/a*.
- Sobre las imágenes de la actividad anterior, añadir un fondo a dicha página web de la misma forma que se ha realizado sobre la Práctica 11.11. Podemos buscar en la Web dicho fondo o utilizar alguno del sistema operativo.

Si en lugar de una imagen de fondo queremos que sea un color (ya vimos la tabla de colores en un apartado anterior), tenemos que utilizar el atributo BGCOLOR en la etiqueta <BODY>. Por lo tanto, la cosa quedaría tal que así:

```
<BODY BGCOLOR="YELLOW">
```

Modificando la Práctica anterior, añadiendo el atributo BGCOLOR a <BODY>, nos quedaría una práctica como el que sigue:

PRÁCTICA 11.11



```
<HTML>
<HEAD>
<TITLE> Decimosegunda p gina Web </TITLE>
</HEAD>
<BODY BGCOLOR="YELLOW">
<H1> Decimosegunda p gina Web</H1>
<BR>
<BR>
<IMG SRC="IMG1.GIF" WIDTH="280" HEIGHT="210" BORDER="3">
<BR>
<BR>
<IMG SRC="IMG1.GIF" WIDTH="680" HEIGHT="210" BORDER="0">
</BODY>
</HTML>
```

ACTIVIDADES 11.16



➤ Nuestra tarea n mero 16 debe tener:

- El t tulo de la p gina debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *pr ctica 16 de nombre y apellidos del alumno/a*.
- Sobre las im genes de la Actividad 11.14, a adir un fondo a dicha p gina web, dicho fondo ser  un color s lido (podemos elegir el color que m s nos guste).

11.1.8 TABLAS

Las tablas nos permiten representar y ordenar cualquier elemento de nuestra presentaci n en diferentes filas y columnas de modo que podamos resumir grandes cantidades de informaci n de una manera que puede representarse r pida y f cil.

Una tabla va siempre delimitada por las etiquetas <table> y </table>. Dentro de una tabla vamos a distinguir dos elementos: filas y columnas:

- Las filas se identifican mediante las etiquetas <tr> y </tr>.
- Las columnas se identifican mediante las etiquetas <td> y </td>.

Hay unas columnas especiales que sirven para el encabezamiento de la tabla identificadas mediante las etiquetas <th> y </th>.

Vamos a ver en la siguiente pr ctica una tabla que tiene 3 filas y 3 columnas: la primera fila contendr  el t tulo de los campos de la tabla.

PRÁCTICA 11.12



```

<HTML>
<HEAD>
<TITLE>Web</TITLE>
</HEAD>
<BODY>
<H1>Tablas básicas</H1>
<TABLE BORDER="1">
<TR>
  <TH>Cabecera 1</TH>
  <TH>Cabecera 2</TH>
  <TH>Cabecera 3</TH>
</TR>
<TR>
  <TD>fila 1</TD>
  <TD>fila 2</TD>
  <TD>fila 3</TD>
</TR>
<TR>
  <TD>fila 1</TD>
  <TD>fila 2</TD>
  <TD>fila 3</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Tablas básicas		
Cabecera 1	Cabecera 2	Cabecera 3
fila 1	fila 2	fila 3
fila 1	fila 2	fila 3

Figura 11.11

Dentro del código anterior tenemos un atributo no visto hasta ahora `BORDER="número"`. Este atributo nos indica el grosor del borde que va a tener nuestra tabla. El valor mínimo es 0, es decir, sin borde, podemos aumentar el valor para ver cómo irá quedando la tabla con unos valores u otros.

ACTIVIDADES 11.17

» Nuestra tarea número 17 debe tener:

- El título de la página debe ser el nombre y apellidos del alumno/a.
- El texto con encabezado1 (<h1>) debe ser *práctica 17 de nombre y apellidos del alumno/a*.
- En esta actividad vamos a realizar la tabla de multiplicar del número 2, empezando por el 1 y terminando por el 10.

Aquí tenemos otra práctica de varias tablas en las que vamos variando el número de filas y columnas y el grosor del borde de la tabla. Tenemos 3 tablas, cada una de tamaño y grosor distinto:

PRÁCTICA 11.13

```
<HTML>
<HEAD>
<TITLE>Web</TITLE>
</HEAD>
<BODY>
<H1>Tablas b&aacute;sicas 2</H1>
<TABLE BORDER="1">
<TR>
  <TH>Cabecera 1</TH>
  <TH>Cabecera 2</TH>
</TR>
<TR>
  <TD>fila 1</TD>
  <TD>fila 2</TD>
</TR>
<TR>
  <TD>fila 1</TD>
  <TD>fila 2</TD>
</TR>
</TABLE>
<BR>
<TABLE BORDER="0">
<TR>
  <TH>Cabecera 1</TH>
  <TH>Cabecera 2</TH>
</TR>
<TR>
  <TD>fila 1</TD>
  <TD>fila 2</TD>
</TR>
</TABLE>
<BR>
<TABLE BORDER="4">
```

```

<TR> <TH>Cabecera 1</TH> </TR>
<TR> <TD>fila 1</TD> </TR>
<TR> <TD>fila 1</TD> </TR>
</TABLE>
</BODY>
</HTML>

```

Tablas básicas 2

Cabecera 1	Cabecera 2
fila 1	fila 2
fila 1	fila 2

Cabecera 1 Cabecera 2
fila 1 fila 2

Cabecera 1
fila 1
fila 1

Figura 11.12

Además de los atributos específicos de cada celda o línea, las tablas pueden ser adicionalmente formateadas a partir de los atributos que nos ofrece la propia etiqueta `<TABLE>`. Los atributos principales son:

- **ALIGN**: alinea horizontalmente la tabla con respecto a su entorno.
- **BACKGROUND**: nos permite colocar un fondo para la tabla a partir de un enlace a una imagen.
- **BGCOLOR**: da color de fondo a la tabla.
- **BORDER**: define el número de puntos del borde principal.
- **BORDERCOLOR**: define el color del borde.
- **CELLPADDING**: define, en puntos, el espacio entre los bordes de la celda y el contenido de la misma.
- **CELLSPACING**: define el espacio entre los bordes (en puntos).
- **HEIGHT**: define la altura de la tabla en puntos o porcentaje.
- **WIDTH**: define la anchura de la tabla en puntos o porcentaje.

Aplicando estos atributos, hay que tener en cuenta que puede aparecer más de uno afectando a una etiqueta, podemos hacer una práctica como la que sigue:

PRÁCTICA 11.14



```
<HTML>
<HEAD>
<TITLE>CASO PRÁCTICO</TITLE>
</HEAD>
<BODY>
<TABLE ALIGN="CENTER" CELSPACING="2" CELLPADDING="2" BORDER="1" BGCOLOR=DDDDDD>
<TR>
<TH COLSPAN="4" ALIGN="CENTER" BGCOLOR="666666"><FONT COLOR="#FFFFFF">Tabla de
muestra</FONT></TD>
</TR>
<TR BGCOLOR="AAAAAA">
<TD>NOMBRE</TD>
<TD ALIGN="CENTER">dato1</TD>
<TD ALIGN="CENTER">dato2</TD>
<TD ALIGN="CENTER"></TD>
</TR>
<TR>
<TD>BALLENA</TD>
<TD ALIGN="CENTER">6000</TD>
<TD ALIGN="CENTER">4000</TD>
<TD ALIGN="CENTER">1500</TD>
</TR>
<TR>
<TD>OSO PARDO</TD>
<TD ALIGN="CENTER">50</TD>
<TD ROWSPAN="2" COLSPAN="2" ALIGN="CENTER" BGCOLOR="RED">0</TD>
</TR>
<TR>
<TD>LINCE</TD>
<TD ALIGN="CENTER">10</TD>
</TR>
<TR>
<TD>TIGRE</TD>
<TD ALIGN="CENTER">300</TD>
<TD COLSPAN="2" ALIGN="CENTER">210</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

11/11/2019 10:20:20 AM C:\Program Files\Internet Explorer\iexplore.exe

Tabla de muestra			
NOMBRE	dato1	dato2	
BALLENA	6000	4000	1500
OSO PARDO	50	0	
LINCE	10		
TIGRE	300	210	

Figura 11.13

ACTIVIDADES 11.18



- Creación de una tabla periódica utilizando tablas. En esta tarea vamos a elaborar una tabla periódica como la siguiente:

[illegible]

Figura 11.14. HTML

ACTIVIDADES 11.19



- » Nuestra tarea número 19 debe tener:
- El título de la página debe ser el nombre y apellidos del alumno/a.
 - El texto con encabezado1 (<h1>) debe ser *práctica 19 de nombre y apellidos del alumno/a*.
 - En esta práctica vamos a realizar la tabla de conversiones como la que viene a continuación.



Para realizar esta práctica necesitaremos utilizar superíndices. Para ello nos valdremos de las etiquetas `<sup>` y `</sup>`. Es decir, para poner 10^{24} , debemos poner `10<sup>24</sup>`.

Factor	Name	Symbol	Factor	Name	Symbol
10^{24}	yotta	Y	10^{-1}	deci	d
10^{21}	zetta	Z	10^{-2}	centi	c
10^{18}	exa	E	10^{-3}	milli	m
10^{15}	peta	P	10^{-6}	micro	μ
10^{12}	tera	T	10^{-9}	nano	n
10^9	giga	G	10^{-12}	pico	p
10^6	mega	M	10^{-15}	femto	f
10^3	kilo	k	10^{-18}	atto	a
10^2	hecto	h	10^{-21}	zepto	z
10^1	deka	da	10^{-24}	yocto	y

Figura 11.15. HTML

11.1.9 MARCOS (FRAMES)

Los *frames*, que significan en castellano marcos, son una manera de partir la página en distintos espacios independientes los unos de los otros, de modo que en cada espacio se coloca una página distinta y por tanto un fichero HTML distinto.

Los *frames* se definen en una página especial en la que el código solo contiene la definición de los *frames* o divisiones y la página que se va a cargar dentro de cada uno de ellos, por lo que deberemos cada una de las páginas que van a contener los marcos de manera independiente. En la página donde definimos los *frames* diremos el número de los mismos que queremos que haya, su tamaño y la página que van a contener.

La principal deferencia que va a haber a la hora de crear una página de *frames* es que en vez de utilizar la etiqueta <BODY>, que sirve normalmente para delimitar lo que se va a ver en la pantalla, se hace uso de la etiqueta <FRAMESET>. Sus principales atributos o parámetros son:

COLS= “ x, y, z,...”, para definir el nº de columnas o marcos verticales que va a tener la ventana, y donde las variables x, y, z... van a definir el tamaño de cada marco vertical, pudiendo definirse en píxeles o en porcentaje. También podemos fijar el tamaño de una columna o columnas como *, y en este caso el tamaño de esta será todo aquel que quede después de definir mediante puntos o porcentajes las demás columnas.

Si definimos tamaños en píxeles, estos serán absolutos, En este caso si todas las *frames* se indican de este modo, los valores se ajustarán para que las *frames* ocupen la totalidad del espacio de la ventana del navegador, no guardando siempre la proporción con la que se definen las *frames*.

Por el contrario, si empleamos medidas en porcentaje, en este caso si los porcentajes suman un valor distinto del 100%, se ajustarán para que coincidan con el tamaño de la ventana. Se podrá combinar con el apartado anterior de forma que algunas *frames* se definan en valor absoluto y otras en porcentaje.

Posibles combinaciones serían:

- ✓ COLS="100,700"...que nos dividiría la ventana en dos columnas, una de 100 puntos y la otra de 700.
- ✓ COLS="10%,*"...que nos daría una columna con el 10% de la ventana y otra con el resto.
- ✓ COLS="20%,*,*"...obteniendo con ello una columna del 20% de la ventana y otras dos que se repartirían por igual el tamaño restante.
- ✓ COLS="*,2*"...que nos creará dos columnas, teniendo la segunda el doble de ancho que la primera, es decir, la primera ocuparía el 33% de la ventana y la segunda el 66% restante.

ROWS=“ x,y,z,...”, para definir el nº de filas o marcos horizontales que va a tener la ventana, y donde las variables x, y y z van a definir el tamaño de cada marco horizontal, pudiendo definirse en píxeles o en porcentaje. También podemos fijar el tamaño de una fila o filas como *, y en este caso el tamaño de esta será todo aquel que quede después de definir mediante píxeles o porcentajes las demás filas. Las combinaciones posibles son muchas, siendo su estructura análoga a las vistas para COLS.

Si definimos tamaños en píxeles, estos serán absolutos, En este caso, si todas los *frames* se indican de este modo, los valores se ajustarán para que las *frames* ocupen la totalidad del espacio de la ventana del navegador, no guardando siempre la proporción con la que se definen los *frames*.

Por el contrario, si empleamos medidas en porcentaje, en este caso si los porcentajes suman un valor distinto del 100%, se ajustarán para que coincidan con el tamaño de la ventana. Se podrá combinar con el apartado anterior de forma que algunos *frames* se definan en valor absoluto y otras en porcentaje.

FRAMEBORDER= “ 0 / 1 / no / yes”, que nos permite definir si los marcos creados van a tener borde o no. Los valores 0 y no son equivalentes, eliminando los bordes, así como 1 y yes, que los muestran. Su valor por defecto es yes / 1, por lo que si no especificamos nada se verán los bordes de todos los marcos.

BORDER= “n”, con n = n° entero = n° de píxeles. En el caso de que se haya establecido **FRAMEBORDER=“1”** o no se haya especificado, este atributo nos permite definir el grosor del borde.

FRAMESPACING= “n”, con n = n° entero = n° de píxeles. Nos permite definir el espacio entre los marcos, y en el caso de que se haya establecido **FRAMEBORDER=“1”** o no se haya especificado, este atributo nos permite definir el grosor del borde.

BORDERCOLOR= “color”, donde color puede venir especificado mediante su nombre web en inglés o mediante su código hexadecimal. Nos permite establecer el color de los bordes de los marcos.

NORESIZE: indica que el *frame* no se puede redimensionar. Si no se pone este atributo colocando el cursor en el borde del *frame*, permitiría su deslizamiento.

Se utiliza la etiqueta **<NOFRAMES>** para mostrar contenido en navegadores que no soporten *frames* (hoy en día casi todos los navegadores soportan esta técnica). Entre la etiqueta **<NOFRAMES>** y su cierre **</NOFRAMES>** se inserta el contenido HTML que deseemos y solo será interpretado por aquellos navegadores que no soporten *frames*. Veamos una práctica con *frames*:

PRÁCTICA 11.15

```
<HTML>
<HEAD>
<TITLE>FRAMES
</TITLE>
</HEAD>
<FRAMESET COLS="30%,70%">
<FRAME SRC="a.html">
<FRAME SRC="b.html" NAME="PRINCIPAL">
</FRAMESET>
<NOFRAMES>
SU NAVEGADOR NO SOPORTA FRAMES, POR FAVOR ACTUALÍCELO A UNA VERSIÓN MÁS RECIENTE.
</NOFRAMES>
</HTML>
```



Figura 11.16

- Como vemos aparece la ventana dividida en 2 partes: en la izquierda iría la página "a.html". Como no la hemos creado nos aparece el mensaje "Archivo no encontrado".

En la ventana de la derecha sucede exactamente lo mismo, al no existir “b.html” nos muestra el mensaje de que no encuentra el archivo. Lo más interesante de los *frames* es la posibilidad que tienen de interactuar entre sí, es decir pulsar un enlace en el *frame* 1 y cargar el contenido en el *frame* 2. Para conseguir esto hay que darle un nombre a los *frames* y luego indicar en el enlace donde se va a cargar mediante el atributo TARGET. Veámoslo:

PRÁCTICA 11.16



```
<HTML>
<HEAD>
<TITLE>Frames</TITLE>
</HEAD>
<FRAMESET COLS=150,*>
<FRAME SRC="frames4.html" NAME=margen>
<FRAME SRC="frames5.html" NAME=principal>
</FRAMESET>
</HTML>
```

➤ Los enlaces de la página *frames4.html*, que es la que se carga en el margen izquierdo, se escriben de la forma:

```
<A HREF="frames1.htm" TARGET="principal"> Frame 1 </A>
```

➤ Hay ciertos valores reservados para TARGET estos son:

- TARGET="_top", hace que la página se cargue en la ventana completa del navegador.
- TARGET="_self", hace que la página se cargue en la misma ventana del *frame* actual.
- TARGET="_parent", hace que la página se cargue en el *frame* “padre”, del que desciende el actual.
- TARGET="_blank", hace que la página se cargue en una nueva ventana.
- TARGET="nombre", hace que la página se cargue en el *frame* llamado *nombre*. Si no existe se carga en una ventana nueva.

```
<A HREF="http://www.google.es" TARGET="_blank"> Abrir Google en una ventana nueva</A>
```

11.1.10 FORMULARIOS

Los formularios son utilizados para realizar búsquedas o bien para introducir datos personales, por ejemplo, en sitios de comercio electrónico. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Los formularios son definidos por medio de las etiquetas <form> y </form>. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario. Dentro de esta etiqueta <form> debemos especificar algunos atributos:

ACTION: define el tipo de acción a llevar a cabo con el formulario. Como ya hemos dicho, existen dos posibilidades:

- El formulario es enviado a una dirección de correo electrónico. el contenido del formulario es enviado a la dirección de correo electrónico especificada por medio de una sintaxis de este tipo:

```
<FORM ACTION= "MAILTO:direccion@correo.com" ...>
```

- El formulario es enviado a un programa o *script* que procesa su contenido. Si lo que queremos es que el formulario sea procesado por un programa, hemos de especificar la dirección del archivo que contiene dicho programa. La etiqueta quedaría en este caso de la siguiente forma:

```
<FORM ACTION="dirección del archivo" ...>
```

La forma en la que se expresa la localización del archivo que contiene el programa es la misma que la vista para los enlaces.

METHOD: este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar este atributo son POST y GET. A efectos prácticos y, salvo que se os diga lo contrario, daremos siempre el valor POST.

ENCTYPE: se utiliza para indicar la forma en la que viajará la información que se mande por el formulario. En el caso más corriente, enviar el formulario por correo electrónico. El valor de este atributo debe de ser "text/plain". Así conseguimos que se envíe el contenido del formulario como texto plano dentro del *email*.

Si queremos que el formulario se procese automáticamente por un programa, generalmente no utilizaremos este atributo, de modo que tome su valor por defecto, es decir, no incluiremos ENCTYPE dentro de la etiqueta <FORM>.

NAME: es el nombre que identifica al formulario, útil si se usan *scripts* dentro de la página.

Cajas de texto. Los controles o cajas de texto permiten al usuario escribir texto en el formulario. Pueden ser cajas de una sola línea (TEXTBOX) o cuadros de texto con varias líneas (TEXTAREA).

Para los TEXTBOX se usa la etiqueta <INPUT TYPE> y pueden ser de texto simple (TEXT) (se ve lo que escribimos) o de tipo contraseña (PASSWORD) (lo que escribimos es visto como una serie de asteriscos). El formato de cada una de ellas es el siguiente:

```
<INPUT TYPE="TEXT">
<INPUT TYPE="PASSWORD">
```

Los atributos que pueden tener tanto TEXT como PASSWORD son los siguientes:

- **SIZE:** tamaño en columnas de la caja de texto.
- **MAXLENGTH:** número máximo de caracteres que se pueden teclear.
- **VALUE:** texto por defecto que aparecerá en el campo.
- **NAME:** el nombre que tendrá el campo de texto y que nos servirá para diferenciarlo de otros campos del mismo tipo.

Aquí tenemos unos ejemplos aplicando los atributos anteriores:

```
<INPUT TYPE="TEXT" NAME="HOJA1" MAXLENGTH=20 SIZE=25>
<INPUT TYPE="PASSWORD" NAME="HOJA2" MAXLENGTH=5 SIZE=15>
```

Usaremos los TEXTAREA si necesitamos que el usuario escriba más líneas de texto. Su sintaxis es la siguiente:

```
<TEXTAREA>Introduzca sus comentarios </TEXTAREA>
```

Los atributos que pueden tener <TEXTAREA> son los siguientes:

- ROWS: filas que ocupará la caja de texto.
- COLS: columnas que ocupará la caja de texto.
- NAME: el nombre del TEXTAREA para referenciarlo posteriormente.

```
<TEXTAREA NAME="hechos" COLS="100" ROWS="10"> </TEXTAREA>
```

Opciones. En primer lugar vamos el tipo RADIO, nos da una serie de opciones en la que solo podemos elegir una de ellas (el marcar una opción implica desmarcar la ya marcada). En este caso su sintaxis es:

```
<INPUT TYPE="RADIO">texto
```

El texto que aparece a continuación del INPUT TYPE es el texto que vera el usuario a la hora de elegir una respuesta u otra. Los atributos que pueden tener RADIO son los siguientes:

- NAME: en este caso el mismo nombre en los campos de selección indica que son respuestas alternativas de la misma pregunta.
- VALUE: el valor de radio seleccionado por el usuario.
- CHECKED: opción activada por defecto (solo puede haber un CHECKED por RADIO).

```
<INPUT TYPE="RADIO" NAME="ELEGIR" VALUE="SI">Respuesta SI
<INPUT TYPE="RADIO" NAME="ELEGIR" VALUE="NO">Respuesta NO
<INPUT TYPE="RADIO" NAME="ELEGIR" VALUE="NS">Respuesta NS/NC
```

Lista de opciones. Podemos presentar la información de manera de lista desplegable, donde al desplegar esa lista con el ratón nos da todas las posibilidades que permite esa selección. Para esto utilizamos <SELECT> y </SELECT>, y las opciones distintas irán entre <OPTION> y </OPTION>.

```
<SELECT NAME="Idioma">
<OPTION>Español</OPTION>
<OPTION>Inglés</OPTION>
<OPTION>Francés</OPTION>
<OPTION>Alemán</OPTION>
</SELECT>
```

Los atributos que pueden tener SELECT son los siguientes:

- NAME: igual que en casos anteriores.
- SIZE: número de opciones que podemos ver a la vez, si es mayor de 1 veremos la lista como un *scroll* y si no lo veremos como una lista desplegable.
- MULTIPLE: significa que podemos seleccionar más de un objeto con la tecla Ctrl (CONTROL) y pulsando con el ratón.

Los atributos que pueden tener OPTION son los siguientes:

- VALUE: igual que para casos anteriores, el valor que realmente tomará el SELECT.
- SELECTED: opción que aparecerá seleccionada por defecto.

Marcas de verificación. Son controles parecidos a los campos de opciones pero que no van agrupados, o sea no son excluyentes. Sirven para que el usuario coloque una marca de sí o no (verdadero o falso, marcado o no marcado). La etiqueta utilizada en este caso es CHECKBOX. Su sintaxis es la siguiente:

```
<P>GUSTOS DEPORTIVOS
<BR>
<INPUT TYPE="CHECKBOX" NAME="OPINION1"> FUTBOL
<INPUT TYPE="CHECKBOX" NAME="OPINION2"> TENIS
<INPUT TYPE="CHECKBOX" NAME="OPINION3" CHECKED> BALONCESTO
```

Como puede apreciarse, el valor NAME de ellos es distinto, a diferencia de RADIO, ya que en RADIO solo era posible una opción y en este caso es posible marcar todas las opciones. Si queremos que el control esté activado por defecto le añadiremos el parámetro CHECKED.

Campos ocultos. Mediante un formulario podemos también enviar datos al programa encargado de procesarlo. Esto se realiza mediante los llamados campos ocultos, que son campos INPUT de tipo HIDDEN y se llaman así porque no se ven en la página web, o sea, el usuario no los ve a menos que se meta en la vista de código de la página. Digo esto porque no vayamos a pensar que con estos campos podemos pasar información privada mediante los formularios. El código para este tipo de campos es algo como esto:

```
<INPUT TYPE=HIDDEN NAME="Curso" VALUE="HTML">
```

Botones del formulario. La parte final del formulario la ocupan los botones. Estos son principalmente 2: el botón **Enviar** y el botón **Limpiar** (borrar todos los datos que hemos introducido en el formulario). En este caso corresponde con <INPUT TYPE="SUBMIT"> (enviar) y con <INPUT TYPE="RESET"> (limpiar información). Así, ambos pueden tener un atributo VALUE que será el texto que aparece dentro del botón.

```
<INPUT TYPE="SUBMIT" VALUE="ENVIAR">
<INPUT TYPE="RESET" VALUE="BORRAR">
```

En la siguiente práctica veremos un formulario completo que enviará los datos por correo electrónico.

PRÁCTICA 11.17



```
<HTML>
<HEAD>
<TITLE> CASO PRÁCTICO</TITLE>
</HEAD>
<BODY>
<FORM OPTION="MAILTO:YYYYY@YYYYY.YYY" ENCTYPE="TEXT/PLAIN">
<B> NOMBRE DE USUARIO: </B> <INPUT TYPE="TEXT" VALUE="ESCRIBE TU NOMBRE...">
<BR>
<B> E-MAIL: </B> <INPUT TYPE="TEXT">
<BR>
<B> CONTRASEÑA: <B> <INPUT TYPE="PASSWORD">
<BR>
<P> <CENTER><B> CONTESTA CORRECTAMENTE LAS SIGUIENTES PREGUNTAS: </B></CENTER> </P>
<P> QUE ES HTML, QUE SIGNIFICA Y QUE TIPO DE LENGUAJE ES? </P>
<TEXTAREA ROWS="5" COLS="90" VALUE="REDACTA CORRECTAMENTE TU RESPUESTAS...">
```

```

REDACTA CORRECTAMENTE TU RESPUESTAS...
</TEXTAREA>
<BR>
<P>
CUAL ES LA MANERA CORRECTA DE PASAR DE OTRA LINEA VALIDADO EN XHTML? </P>
<INPUT TYPE="CIRCLE"> BR
<BR>
<INPUT TYPE="CIRCLE"> PASARLINEA;
<BR>
<INPUT TYPE="CIRCLE"> &lt;BR&gt;
<BR>
<INPUT TYPE="CIRCLE"> &lt;BR/&gt;
<BR>
<BR>
<P> CUALES DE ESTAS ETIQUETAS SE USAN ACTUALMENTE? </P>
<INPUT TYPE="CHECKBOX"> &lt;BR&gt;
<BR>
<INPUT TYPE="CHECKBOX"> &lt;SET&gt;
<BR>
<INPUT TYPE="CHECKBOX"> &lt;P&gt;
<BR>
<INPUT TYPE="CHECKBOX"> &lt;GO&gt;
<BR>
<INPUT TYPE="CHECKBOX"> &lt;TABLE&gt;
<BR>
<P> QUE TYPE="XXXX" TIENE QUE PONERSE DENTRO DEL INPUT PARA QUE NO SE PUEDA VER LO
QUE SE ESCRIBE?
<BR>
<INPUT TYPE="PASSWORD" VALUE="A">
<BR>
<BR>
<INPUT TYPE="SUBMIT" VALUE="ENVIAR">
</BODY>
</HTML>

```

ACTIVIDADES 11.20



➤ Hacer un formulario que envíe por correo electrónico la siguiente información:

- Nombre.
- Apellidos.
- Dirección.
- Correo electrónico.
- Nombre de usuario.
- Contraseña.
- Sexo (hombre, mujer, ns/nc).
- Profesión (estudiante, trabajador, desempleado).
- Obtener información de: viajes, trabajo, informática, telefonía, deportes.

11.2 MODIFICACIÓN DE LA APARIENCIA DE UN DOCUMENTO WEB CON HOJAS DE ESTILOS

El lenguaje HTML está limitado a la hora de aplicarle forma a un documento. Esto es así porque fue concebido para otros usos (científicos sobre todo), distinto a los actuales, mucho más amplios.

La solución a este problema reside en la separación por un lado de la información y, por otro lado, el formateo de dicha información, lo que en la práctica se traduce en la adopción de las Cascading Style Sheets, “hojas de estilo en cascada”, llamadas también CSS.

Con la expresión “en cascada” (*cascading*) se hace referencia a una de las características principales de esta tecnología, por la que es posible incorporar en el documento distintas hojas de estilo, cada una de las cuales, según una serie de reglas jerárquicas, prevalece sobre las otras.

Las CSS fueron introducidas por Microsoft, a partir de la tercera versión de Internet Explorer, y son soportadas parcialmente por Netscape solo desde la cuarta versión del homónimo navegador. Quien accede con un navegador obsoleto a un documento formateado con CSS, se encontrará ante una página gestionada por las opciones por defecto del navegador (el fondo, por ejemplo, será gris; el tipo de texto será “Times new roman”, etc).

Más ventajas, ya que si con el HTML tan solo podíamos definir atributos en las páginas con píxeles y porcentajes, ahora podemos definir utilizando muchas más unidades como:

- Píxeles (px) y porcentaje (%), como antes.
- Pulgadas (in).
- Puntos (pt).
- Centímetros (cm).

A estas unidades se les conoce como unidades CSS y las vamos a utilizar mucho de ahora en adelante.

11.2.1 USOS DE LAS CSS

- **Pequeñas partes de la página.** Para definir estilos en secciones reducidas de una página se utiliza la etiqueta . Con su atributo *style* indicamos en sintaxis CSS las características de estilos. Lo vemos con un ejemplo, pondremos un párrafo en el que determinadas palabras las vamos a visualizar en color verde.

```
...
<div style="....."> ... </span>
...
```

- **Estilo definido para una etiqueta.** De este modo podemos hacer que toda una etiqueta muestre un estilo determinado. Por ejemplo, podemos definir un párrafo entero en color rojo y otro en color azul. Para ello utilizamos el atributo *style*, que es admitido por todas las etiquetas del HTML.

```
...
<p style="....."> ... </p>
...
```

- **Estilo definido en una parte de la página.** Con la etiqueta <DIV> podemos definir secciones de una página y aplicarle estilos con el atributo *style*, es decir, podemos definir estilos de una vez a todo un bloque de la página.

```
...
<div style=".....">
...
</div>
```

- **Estilo definido para toda una página.** Podemos definir, en la cabecera del documento, estilos para que sean aplicados a toda la página. Dentro de la etiqueta <HEAD> deben ir definidos los estilos.

```
<head>
...
<style type="text/css">
...
... aquí irían definidos los estilos
...
</style>
</head>
```

- **Estilo definido para todo un sitio web.** Podemos definir los estilos de todo un sitio web. Esto se consigue creando un archivo externo con extensión css que contendrá las declaraciones de estilos de la página. Debemos enlazar cada una de las páginas de mi sitio a ese archivo externo CSS. Para hacer referencia dentro de <HEAD> iría la referencia al archivo externo.

```
<head>
...
<link rel="stylesheet" type="text/css" href="web.css">
...
</head>
```

11.2.2 REGLAS DE PRECEDENCIA EN LOS ESTILOS

En muchas ocasiones más de una declaración de estilos afecta a la misma porción de la página. A continuación se presentan de menor a mayor precedencia las declaraciones:

- ✓ Declaración de estilos con fichero externo (para todo un sitio web).
- ✓ Declaración de estilos para toda la página (con la etiqueta <STYLE> en la cabecera de la página).
- ✓ Estilos definidos en una parte de la página (con la etiqueta <DIV>).
- ✓ Definidos en una etiqueta en concreto (utilizando el atributo style en la etiqueta en cuestión).
- ✓ Declaración de estilo para una porción pequeña del documento (con la etiqueta).

11.2.3 SINTAXIS DE LAS HOJAS DE ESTILO EN CASCADA

Para definir un estilo se utilizan atributos seguidos de : y el valor que le deseemos asignar terminando en:

```
Estilo: atributos;
```

Para definir el estilo de una etiqueta se escribe la etiqueta seguida de la lista de atributos encerrados entre llaves.

11.2.4 ATRIBUTOS DE LAS HOJAS DE ESTILO

Atributos de estilo para el texto

- **color:** indica el color del texto. Los valores pueden ser el nombre del color en inglés o el valor RGB de dicho color empezando por #.
- **font-size:** tamaño de la fuente. El valor de la fuente se puede establecer mediante unidades CSS o con las atributos: *xx-small*, *x-small*, *small*, *médium*, *large*, *x-large*, *xx-large*.
- **font-family:** tipo de fuente. El valor será el del tipo de fuente que queramos utilizar: Times, Arial, Verdana, etc.
- **font-weight:** anchura de los caracteres. El valor vendrá determinado por los atributos: *normal*, *bold*, *bolder*, *lighter*, 100, 200, 300, 400, 500, 600, 700, 800, 900.
- **font-style:** es el estilo de la fuente, que puede ser normal, itálica u oblicua. Los valores son: *normal*, *italic*, *oblique*.

Vamos a ver en la siguiente práctica la utilización de fuentes en CSS.

PRÁCTICA 11.18



En primer lugar el archivo con la declaración de estilos "estilos.css".

```
P {
  font-size: 12pt;
  font-family: arial,helvetica;
  font-weight: bolder;
  font-style: italic;
}
H1 {
  color: orange;
  font-size: 36pt;
  font-family: verdana,arial;
}
TD {
  color: blue;
  font-size: 11pt;
  font-family: verdana,arial;
}
BODY {
  font-family: arial;
  color: Black;
  font-style: normal;
}
```

Y ahora el fichero de prueba de esos estilos "prueba.html".

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="estilos.css">
<title>p&acute;gina que lee estilos</title>
```

```

</head>
<body>
<H1>Página que lee estilos</H1>
<p>Esta página tiene en la cabecera la etiqueta necesaria para enlazar con
la hoja de estilos. Es muy fácil.</p>
Esta página tiene en la cabecera la etiqueta necesaria para enlazar con la
hoja de estilos. Es muy fácil.
<br>
<br>
<table width="300" cellspacing="2" cellpadding="2" border="1">
<tr>
    <td>Esto está dentro de un td, luego tiene estilo propio, declarado en
el fichero externo</td>
</tr>
<tr>
    <td>La segunda fila del TD</td>
</tr>
</table>
</BODY>
</HTML>

```

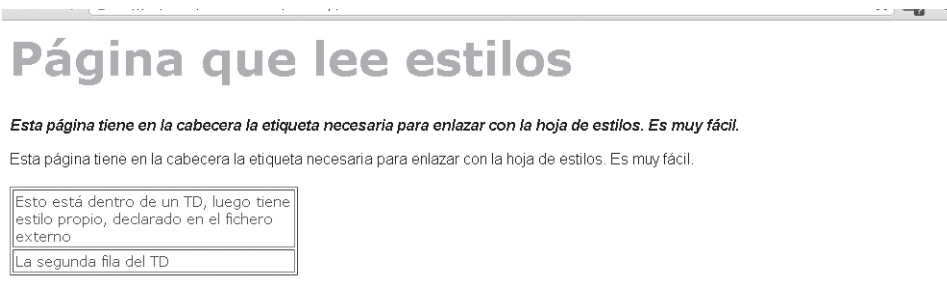


Figura 11.17

ACTIVIDADES 11.21



➤ Haga un archivo CSS con su correspondiente comprobación en un archivo HTML que redefina las etiquetas siguientes:

- **P:** color negro, fuente arial, ancho normal, estilo cursiva, tamaño 12 puntos.
- **TD:** color azul, fuente arial, tamaño 12 puntos, estilo normal, ancho negrita.
- **H1:** color rojo, fuente Times, tamaño 32 puntos, estilo negrita.
- **H2:** color verde, fuente: Verdana, tamaño 24 puntos, estilo negrita.

ACTIVIDADES 11.22



➤ Haga un archivo HTML que lleve integrado el CSS correspondiente que redefina las etiquetas siguientes:

- **P**: color azul, fuente helvética, ancho negrita, estilo normal, tamaño 14 puntos.
- **TD**: color negro, fuente Times, tamaño 11 puntos, estilo itálica.
- **H1**: color verde, fuente Arial, tamaño 48 puntos, estilo negrita.
- **H2**: color rojo, fuente Arial, tamaño 26 puntos, estilo negrita.

Atributos de estilo para párrafo

- **line-height**: el espaciado entre líneas. Para definir el valor utilizaremos las unidades CSS.
- **text-decoration**: decoración del texto. Valores posibles: *none*, *underline*, *overline*, *line-through*. Excepto el *none* el resto pueden usarse a la vez.
- **text-align**: alineación del texto. Posibles valores: *left*, *right*, *center* y *justify*.
- **text-indent**: sangrado o márgenes en las páginas. Los valores son unidades CSS.
- **text-transform**: permiten transformar el texto. Los valores son: *capitalize*, *uppercase*, *lowercase*, *none*.

Vamos a ver la siguiente práctica sobre utilización de párrafos en CSS.

PRÁCTICA 11.19



En primer lugar el archivo con la declaración de estilos "estilos2.css".

```
P {
  line-height: 1cm;
  text-decoration: underline;
  text-align: justify;
  text-transform: capitalize;
  text-indent: 2cm;
}
H1 {
  text-align: right;
  text-transform: uppercase;
}
TD {
  text-decoration : line-through;
  text-transform: lowercase;
}
BODY {
  line-height: 1cm;
  text-decoration: none;
  text-align: left;
  text-transform: lowercase;
  text-indent: 5cm;
}
```

Y ahora el fichero de prueba de esos estilos "prueba2.html".

```
<HTML>
<HEAD>
  <LINK REL="STYLESHEET" TYPE="TEXT/CSS" HREF="estilos2.css">
  <TITLE>P&acute;gina que lee estilos</TITLE>
</HEAD>
<BODY>
<H1>P&acute;gina que lee estilos</H1>
<P> Esta p&acute;gina tiene en la cabecera la etiqueta necesaria para enlazar con
la hoja de estilos. Es muy f&ácil. Esta p&acute;gina tiene en la cabecera la
etiqueta necesaria para enlazar con la hoja de estilos. Es muy f&ácil. Esta
p&acute;gina tiene en la cabecera la etiqueta necesaria
para enlazar con la hoja de estilos. Es muy f&ácil. Esta p&acute;gina tiene
en la cabecera la etiqueta necesaria para enlazar con la hoja de estilos. Es muy
f&ácil.
</P>
<BR>
Esta p&acute;gina tiene en la cabecera la etiqueta necesaria para enlazar con la
hoja de estilos. Es muy f&ácil. Esta p&acute;gina tiene en la cabecera la
etiqueta necesaria para enlazar con la hoja de estilos. Es muy f&ácil. Esta
p&acute;gina tiene en la cabecera la etiqueta necesaria para enlazar con la hoja
de estilos. Es muy f&ácil. Esta p&acute;gina tiene en la cabecera la etiqueta
necesaria para enlazar con la hoja de estilos. Es muy f&ácil.
<BR>
<BR>
<TABLE WIDTH="300" CELSPACING="2" CELLPADDING="2" BORDER="1">
<TR>
  <TD>Esto est&acute; dentro de un TD, luego tiene estilo propio, declarado en
el fichero externo</TD>
</TR>
<TR>
  <TD>La segunda fila del TD</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

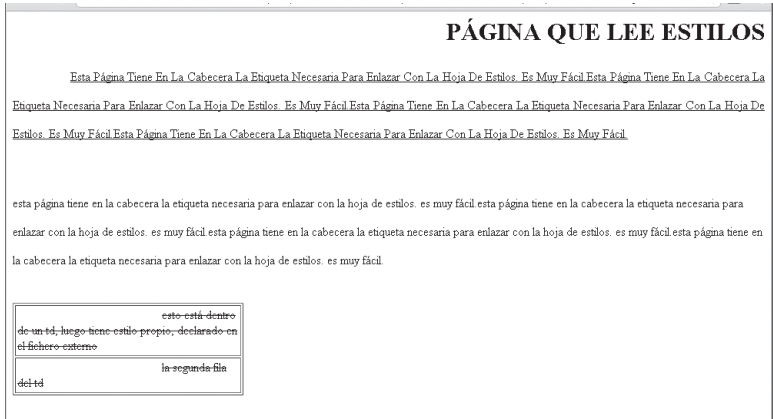


Figura 11.18

ACTIVIDADES 11.23



➤ Hacer un fichero CSS externo y un HTML para comprobarlo:

- **P**: alto de línea: 12 pt, subrayado y sobrerayado, justificado, todo en minúsculas, sangría 3 cm.
- **H1**: alto de línea: 36 pt, tachado, alineado a la derecha, todo en mayúsculas.
- **td**: alto de línea: 24 pt, sobrerayado, centrado, letra capital, sangría 1 cm.
- **body**: alto de línea: 8 pt, sin subrayar, alineación izquierda, todo en minúscula, sangría 2 cm.

ACTIVIDADES 11.24



➤ Hacer un fichero HTML que lleve incorporado el CSS siguiente:

- **P**: alto de línea: 36 pt, tachado, alineado a la derecha, todo en mayúsculas.
- **H1**: alto de línea: 24 pt, subrayado y sobrerayado, justificado, todo en minúsculas, sangría 3 cm.
- **TD**: alto de línea: 8 pt, sin subrayar, alineación izquierda, todo en minúscula, sangría 2 cm.
- **BODY**: alto de línea: 24 pt, sobrerayado, centrado, letra capital, sangría 1 cm.

Atributos para fondo

- **background-color**: color de fondo. Los valores son el color en inglés o el valor RGB del color que queramos precedido por #.
- **background-image**: nombre de la imagen con su camino relativo o absoluto.

Vamos a ver la práctica siguiente sobre la utilización del fondo en CSS.

PRÁCTICA 11.20



En primer lugar el archivo con la declaración de estilos "estilos3.css".

```
P {  
  background-color: orange;  
}  
TABLE {  
  background-color: lightblue;  
}  
BODY {  
  background-image: url(fondo.gif);  
}
```


Y ahora el fichero de prueba de esos estilos "prueba3.html".

```
<HTML>
<HEAD>
  <LINK REL="STYLESHEET" TYPE="TEXT/CSS" HREF="estilos3.css">
  <TITLE>P gina que lee estilos</TITLE>
</HEAD>
<BODY>
<H1>P gina que lee estilos</H1>
<p>Esta p gina tiene en la cabecera la etiqueta necesaria para enlazar con
la hoja de estilos.
Es muy f cil.</p>
<BR>
Esta p gina tiene en la cabecera la etiqueta necesaria para enlazar con la
hoja de estilos.
Es muy f cil.
<BR>
<BR>
<TABLE WIDTH="300" CELSPACING="2" CELLPADDING="2" BORDER="1">
<TR>
  <TD>Esto est  dentro de un TD, luego tiene estilo propio, declarado en
el fichero externo</TD>
</TR>
<TR>
  <TD>La segunda fila del TD</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

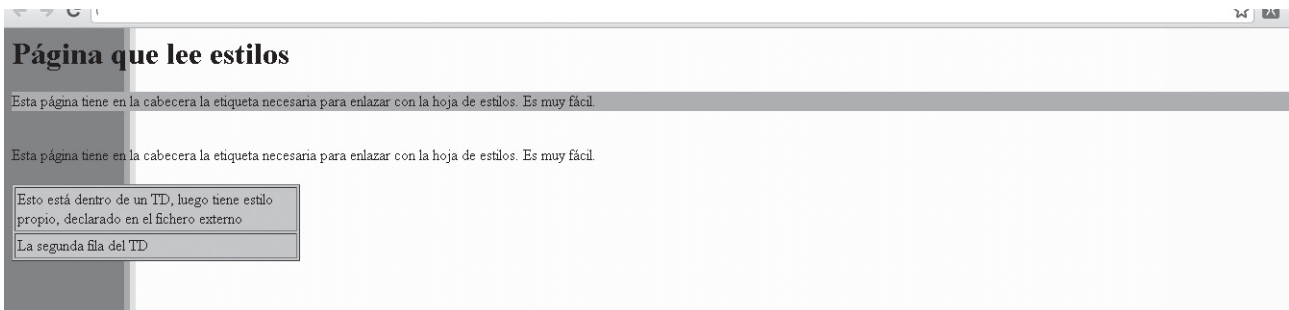


Figura 11.19

ACTIVIDADES 11.25



➤ A ada un fondo a las tareas de las Actividades 11.23 y 11.24.

Atributos para casillas

- **margin-left, margin-right, margin-top, margin-bottom:** margen izquierdo, derecho, superior e inferior respectivamente. Su valor se establece mediante unidades CSS.
- **padding-left, padding-right, padding-top, padding-bottom:** espacio insertado entre el borde y el inicio del texto. Su valor también se mide en unidades CSS.
- **border-color:** color de borde. Igual que el resto de colores vistos hasta ahora.
- **border-style:** estilo del borde. Su valor puede ser: *none, dotted, solid, doublé, groove, ridge, inset, outset*.
- **border-width:** el tamaño del borde del elemento al que lo aplicamos. El valor se expresa en unidades CSS.

Vamos a ver la práctica siguiente sobre utilización de párrafos en CSS.

PRÁCTICA 11.21



En primer lugar el archivo con la declaración de estilos "estilos4.css".

```
P {
border-color:orange;
margin-left: 2cm;
border-style: solid;
border-width: 10px;
}
H1 {
clear: right;
border-color:black;
border-style:dotted;
margin-top: 10px;
}
TABLE {
border-color:red;
margin-left: 6cm;
border-style: double;
border-width: 20px;
}
```

Y ahora el fichero de prueba de esos estilos "prueba4.html".

```
<HTML>
<HEAD>
  <LINK REL="STYLESHEET" TYPE="TEXT/CSS" HREF="estilos4.css">
  <TITLE>P&aacute;gina que lee estilos</TITLE>
</HEAD>
<BODY>
<H1>P&aacute;gina que lee estilos</H1>
<p>Esta p&aacute;gina tiene en la cabecera la etiqueta necesaria para enlazar con
la hoja de estilos.
Es muy f&aacute;cil.</p>
<BR>
```

Esta página tiene en la cabecera la etiqueta necesaria para enlazar con la hoja de estilos.

Es muy fácil.

<TABLE WIDTH="300" CELSPACING="2" CELLPADDING="2" BORDER="1">

<TR>

<TD>Esto está dentro de un TD, luego tiene estilo propio, declarado en el fichero externo</TD>

</TR>

<TR>

<TD>La segunda fila del TD</TD>

</TR>

</TABLE>

</BODY>

</HTML>

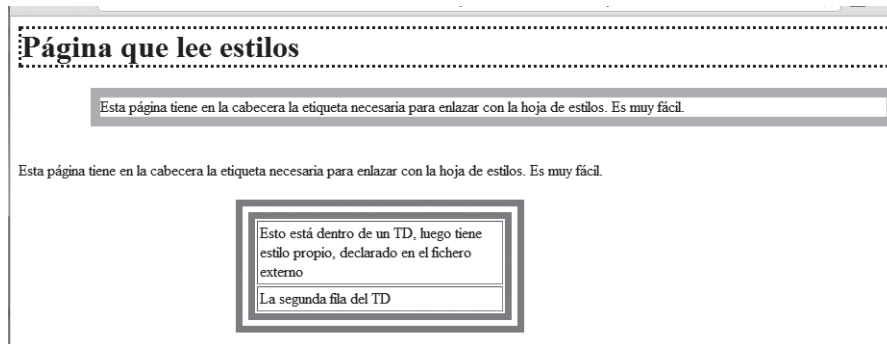


Figura 11.20

ACTIVIDADES 11.26



➤ Hacer un fichero CSS y un fichero de prueba HTML con las siguientes propiedades:

a. TABLE

- márgenes: 1 cm izquierdo, 2 cm derecho, 1 cm superior, 2 cm inferior.
- separación del texto con la tabla: 1 punto izquierdo, 2 punto derecho, 1 punto superior, 2 punto inferior.
- color de borde: azul marino (*navy*).
- estilo del borde: doble.
- ancho del borde: 2 cm.

b. TD

- ancho del borde: 1cm.
- color de borde: azul claro (*cyan*).
- estilo del borde: inset.

ACTIVIDADES 11.27

➤ Hacer un fichero HTML con CSS incorporado con las siguientes propiedades:

- a. TABLE
 - márgenes: 2 cm izquierdo, 2 cm derecho, 2 cm superior, 2 cm inferior.
 - separación del texto con la tabla: 12 puntos izquierdo, 24 puntos derecho, 12 puntos superior, 24 puntos inferior.
 - color de borde: rojo.
 - estilo del borde: *outset*.
 - ancho del borde: 5 cm .
- b. TD
 - ancho del borde: 2 cm.
 - color de borde: rosa (*pink*).
 - estilo del borde: (*double*).

11.2.5 DEFINIR ESTILOS UTILIZANDO CLASES

Las clases nos sirven para crear definiciones de estilos que se pueden utilizar repetidas veces. Una clase se puede definir entre las etiquetas <STYLE> (en la cabecera del documento), o en un archivo externo a la página. Para definirlas utilizamos la siguiente sintaxis, un punto seguido del nombre de la clase y entre llaves los atributos de estilos deseados. De esta manera:

```
.nombredelaclase {
  Atributo1:valor1;
  atributo2:valor2;
  ... }
```

Una vez tenemos una clase, podemos utilizarla en cualquier etiqueta HTML. Para ello utilizaremos el atributo *class*, poniéndole como valor el nombre de la clase (véase Práctica 11.22):

PRÁCTICA 11.22

```
<HTML>
<HEAD>
<TITLE>Ejemplo de la utilizaci&ocirc;n de clases</TITLE>
<STYLE type="text/css">
  .fondonegroletrasblancas
  {
    background-color:black;
    color:white;
    font-size:12;
    font-family:arial
  }
  .letrasverdes
  {
```

```

        color:#009900
    }
</STYLE>
</HEAD>
<BODY>
<H1 CLASS=letrasverdes>Titulo 1</H1>
<H1 CLASS=fondonegroletrasblancas>Titulo 2</H1>
<P CLASS=letrasverdes>
Esto es un párrafo con estilo de letras verdes</P>
<P CLASS=fondonegroletrasblancas>
Esto es un párrafo con estilo de fondo negro y las letras
blancas. Es todo!</P>
</BODY>
</HTML>

```

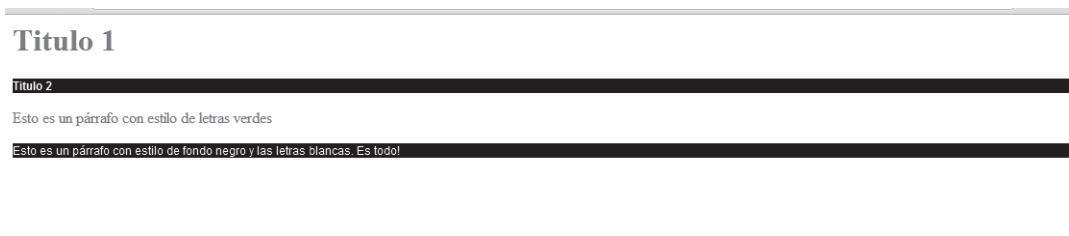


Figura 11.21

11.2.6 ESTILO EN LOS ENLACES

Una técnica muy habitual, que se puede realizar utilizando las hojas de estilo en cascada y no se podía en HTML, es la definición de estilos en los enlaces, quitándoles el subrayado o hacer enlaces en la misma página con distintos colores. Para aplicar estilo a los enlaces debemos definirlos para los distintos tipos de enlaces que existen (visitados, activos...). Utilizaremos la siguiente sintaxis, en la declaración de estilos global de la página (<STYLE>) o del sitio (definición en un archivo externo):

- **Enlaces normales:** A:link {atributos}
- **Enlaces visitados:** A:visited {atributos}
- **Enlaces activos** (los enlaces están activos en el preciso momento en que se pulsa sobre ellos): A:active {atributos}
- **Enlaces hover** (cuando el ratón está encima de ellos): A:hover {atributos}

El atributo para definir enlaces sin subrayado es *text-decoration:none*, y para darles color es *color:tu_color*. También podemos definir el estilo de cada enlace en la propia etiqueta <A>, con el atributo *style*. De esta forma podemos hacer que determinados enlaces de la página se vean de manera distinta.

PRÁCTICA 11.23



```
<HTML>
<HEAD>
  <TITLE>ejemplos de estilo en enlaces</TITLE>
  <STYLE type="text/css">
    A:link {
      text-decoration:none;
      color:#0000cc;
    }
    A:visited {
      text-decoration:none;
      color:#ffcc33;
    }
    A:active {
      text-decoration:none;
      color:#ff0000;
    }
    A:hover {
      text-decoration:underline;
      color:#999999;
      font-weight:bold
    }
  </STYLE>
</HEAD>
<BODY>
<A HREF="http://nofunciona.com">Enlace normal</A>
<BR>
<BR>
<A HREF="enlaces.html">Enlace visitado</A>
Pulsar este enlace para verlo activo, poner el ratón por encima para que
cambie.
</BODY>
</HTML>
```

ACTIVIDADES 11.28



➤ Hacer una página web con las siguientes características:

- Los *links* normales (no visitados), visitados y activos deben tener las siguientes características: deben ser de color negro, no deben ir subrayados, su tamaño será dos veces el tamaño normal de letra en estos casos.
- Cuando pasemos el ratón por encima de alguno de estos *links*, debe aparecer lo siguiente: el color debe ser amarillo, debe aparecer un fondo azul oscuro (por ejemplo: *navy*), el grosor de la letra cambiará a negrita, el tamaño de la letra deberá crecer tres veces el tamaño normal de la letra, deberá aparecer con doble subrayado (por arriba y por abajo).



RESUMEN DEL CAPÍTULO

En este capítulo hemos tratado dos de los lenguajes de programación orientada a clientes más utilizados y populares: el lenguaje HTML y las hojas de estilo en cascada CSS. Se ha comenzado desde el principio, suponiendo que el alumno tiene pocas nociones previas, tanto en HTML como en CSS. Se ha intentado incorporar el máximo número de prácticas y de actividades a su vez para que el alumno/a pueda poner en práctica lo estudiado en el capítulo.



EJERCICIOS PROPUESTOS

- 1. Defina con sus palabras qué es HTML.
- 2. ¿Qué limitaciones tiene HTML con respecto a presentar la información de forma vistosa?
- 3. Defina con sus palabras qué es CSS.
- 4. ¿Qué cree que es anterior, HTML o CSS? ¿Por qué?
- 5. Busque en Internet información sobre HTML5, sus principales novedades y características.
- 6. Busque en Internet información sobre CSS3, sus principales novedades y características.



TEST DE CONOCIMIENTOS



1 La etiqueta que nos permite hacer una tabla es:

- a) `</TABLE>`.
- b) `<TABLE>`.
- c) `<TD>`.
- d) `<TR>`.
- e) Todas son ciertas.
- f) Todas son falsas.

2 Si queremos que el texto en HTML nos salga en negrita, debemos utilizar la etiqueta:

- a) `<BR>`.
- b) `<P>`.
- c) `<N>`.
- d) `<B>`.
- e) `<I>`.

3 Si queremos que el texto en HTML nos salga en cursiva, debemos utilizar la etiqueta:

- a) `<BR>`.
- b) `<P>`.
- c) `<N>`.
- d) `<B>`.
- e) `<I>`.

4 Con CSS para dejar un margen por la derecha entre una tabla y el texto que escribimos debemos utilizar:

- a) *magin-top*.
- b) *padding-top*.
- c) *margin-right*.
- d) *padding-right*.

5 En CSS los enlaces que se activan cuando ponemos el raton sobre ellos se llaman:

- a) `A:link`.
- b) `A:hover`.
- c) `A:active`.
- d) `A:visited`.

12

Programación de documentos web utilizando lenguajes de *script* del cliente

OBJETIVOS DEL CAPÍTULO

- ✓ Diferenciar ejecución web del lado del cliente y del servidor.
- ✓ Introducir el modelo de objetos de documento DOM.
- ✓ Conocer el lenguaje de programación web del lado del cliente: JavaScript.
- ✓ Realizar ejemplos y prácticas sobre dicho lenguaje.

12.1 DIFERENCIAS ENTRE LA EJECUCIÓN EN LADO DEL CLIENTE Y DEL SERVIDOR

El navegador es una aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden.

Cuando nosotros pulsamos sobre un enlace hipertexto, en realidad lo que pasa es que establecemos una petición de un archivo HTML residente en el servidor (un ordenador que se encuentra continuamente conectado a la red) el cual es enviado e interpretado por nuestro navegador (el cliente).

Así pues, podemos hablar de lenguajes de lado servidor, que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no solo se encuentra el HTML, sino también el Java y el JavaScript, los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente “digeridos” por el navegador y no necesitan un pretratamiento.

Cada uno de estos tipos tiene por supuesto sus ventajas y sus inconvenientes. Así, por ejemplo, un lenguaje de lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con *scripts* de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas. Inversamente, un lenguaje de lado servidor es independiente del cliente, por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

Tabla 12.1 Lenguajes del lado del cliente y del servidor

Lenguajes del lado del Cliente	Lenguajes del lado del Servidor	Lenguajes del lado del Cliente y del Servidor
HTML JavaScript Applets de Java Visual Basic Script Flash CSS	CGI PHP ASP PERL JSP	DHTML XML

12.2 MODELO DE OBJETOS DEL DOCUMENTO DOM

12.2.1 INTRODUCCIÓN

El Modelo de Objetos de Documento, o DOM, es la interfaz que le permite acceder y manipular, mediante la programación a los contenidos de una página web (o documento). Proporciona una representación estructurada, orientada a objetos, de los elementos individuales y el contenido de una página, con métodos para recuperar y fijar las

propiedades de dichos objetos. Además, proporciona métodos para agregar y eliminar dichos objetos, permitiéndole crear contenido dinámico.

El DOM proporciona también una interfaz para trabajar con eventos, permitiéndole capturar y responder a las acciones del usuario o del navegador. Esta característica es cubierta someramente en este tutorial, pero los detalles pueden encontrarse en otro artículo. Para este artículo en particular la discusión se centrará en la representación DOM de un documento y los métodos que proporciona para acceder a dichos objetos.

12.2.2 NIVELES DEL DOM Y SOPORTE

Cuando JavaScript fue introducido inicialmente en los navegadores, se requería algún tipo de interfaz para permitir el acceso a los elementos en la página vía *scripting*. Cada vendedor tenía su propia implementación pero emergieron estándares de facto para producir un modelo bastante simple común para todos.

Por ejemplo, la mayor parte de los navegadores usaban un *array* de objetos *Image* para representar todas las etiquetas IMG en una página. Entonces, podía accederse a éstas y manipularlas vía JavaScript. Un simple *rollover* de imágenes usaría un código como éste:

```
document.images[3].src = "graphics/button2.gif"
```

Estos modelos iniciales estaban limitados. Solo proporcionaban acceso a unos pocos tipos de elementos y atributos, como imágenes, *links* o formularios.

Como los vendedores lanzaron nuevas versiones de los navegadores, expandieron dicho modelo. Pero estaban en desacuerdo unos con otros con bastante frecuencia, lo que conducía tanto a problemas de compatibilidad entre los diferentes navegadores, como a los vendedores, tratando de aventajar a los demás añadiendo sus propias características nuevas.

Afortunadamente, la mayor parte de los desarrolladores de navegadores empezaron a adoptar el estándar del DOM fijado por el World Wide Web Consortium, nos referimos en concreto a Internet Explorer, Netscape y Opera.

Con el fin de proporcionar cierta compatibilidad hacia atrás, se definieron diferentes niveles del estándar del DOM. Encontrará referencias al DOM Nivel 0 (o "DOM0"), que corresponde al modelo usado por los primeros navegadores, principalmente Internet Explorer y Netscape, anteriores a la versión 4 de cada uno. A continuación está el DOM1, que fue adoptado en 1998, y que cubre algunas de las características introducidas en la versión 4 de los navegadores.

12.2.3 NAVEGADORES COMPATIBLES CON LOS ESTÁNDARES

La mayoría de los navegadores actuales (a partir de la versión 5 del Internet Explorer, Firefox, Chrome, Opera y Safari) soportan el estándar DOM2, o al menos una buena parte de éste. Además, continúan soportando algunas características de los anteriores niveles del DOM, o sus propias extensiones propietarias, por lo que son compatibles con páginas web antiguas.

12.3 JAVASCRIPT

JavaScript es un lenguaje de programación creado por la empresa Netscape. Es el lenguaje de programación más utilizado en Internet para añadir interactividad a las páginas web. No confundir el JavaScript con el Java. El Java es un lenguaje de programación de propósito general como lo son el C++ o el Visual Basic.

Un programa en JavaScript se integra en una página web (entre el código HTML) y es el navegador el que lo interpreta (ejecuta). Es decir el JavaScript es un lenguaje interpretado, no compilado (no se genera ningún tipo de fichero objeto o *exe*).

Para programar en JavaScript solo necesitamos un editor de texto (podemos utilizar el Bloc de Notas del Windows) y un navegador para ejecutarlo. ¿Por qué el JavaScript y no otro lenguaje de programación? Porque:

- ✓ Es **moderno** (tiene pocos años).
- ✓ Es **sencillo** (su hermano mayor: el Java, es bastante más complejo).
- ✓ Es **útil** (el desarrollo de Internet, ha sido muy rápido en los últimos años).
- ✓ Es **potente**: permite la POO (Programación Orientada a Objetos).
- ✓ Es **barato**: solo necesitamos un editor de textos y un navegador. Permite la “programación visual” (ventanas, botones, colores, formularios, etc.).

12.3.1 CONSIDERACIONES INICIALES

Todo código JavaScript va siempre entre las siguientes etiquetas:

```
<SCRIPT LANGUAGE="JavaScript"> ... </SCRIPT>
```

O simplemente:

```
<SCRIPT>...</SCRIPT>
```

Todas las líneas de código JavaScript acaban siempre en “;” (excepto algunos casos que veremos más adelante). Si queremos incluir comentarios (zonas de texto que solo son visualizadas al ver el código fuente, no serán mostradas durante la ejecución del programa) se utilizará // si el comentario es de una línea y /* ... */ si el comentario ocupa más de una línea. Si el navegador no soporta JavaScript debemos incluir las siguientes etiquetas, que serán ejecutadas por dichos navegadores (los que no soportan JavaScript):

```
<NOSCRIPT>
</NOSCRIPT>
```

12.3.2 PRIMER EJEMPLO

El siguiente código presenta el primer ejemplo de JavaScript, en este caso simplemente mostrará por pantalla un mensaje “Hola Mundo!”.

```
<HTML>
<HEAD>
<TITLE>EJE1.HTML</TITLE>
```

```
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
alert("Hola Mundo!");
</SCRIPT>
</BODY>
</HTML>
```

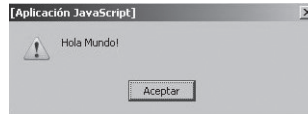


Figura 12.1. Primer ejemplo

Nuestro primer programa JavaScript contiene una única sentencia: `alert("¡Hola Mundo!");` que “abre” una ventana con el mensaje que tenemos entre comillas.

Al hacer clic en **Aceptar** de la ventana “alert”, se acaba el programa JavaScript (se encuentra la etiqueta `</SCRIPT>`) y continua ejecutándose la página HTML.

ACTIVIDADES 12.1



➤ Añadir al ejemplo anterior las siguientes características:

- Después del primer mensaje, se debe mostrar otro que diga “Soy el primer script”.
- Añadir algunos comentarios que expliquen el funcionamiento del código.

12.4 VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT

Una de las grandes aportaciones de JavaScript a la creación de interfaces web es la posibilidad de acceder al contenido de los campos de los formularios para realizar acciones sobre los valores introducidos por el usuario, modificarlos y, en última instancia, validarlos. La validación de los datos de un formulario mediante *scripts* JavaScript no sustituye a la validación que debe realizarse, por motivos de seguridad, en la aplicación del servidor que recibe la información. Sin embargo, al añadir una validación local con JavaScript, la experiencia de usuario mejora notablemente, al no ser necesario enviar los datos al servidor y esperar su respuesta para obtener solo un mensaje informando de la incorrección de la información suministrada. Resulta frustrante cumplimentar un formulario, pulsar el botón Enviar y esperar 30 ó 40 segundos para saber que hemos introducido mal un campo.

JavaScript, desde sus comienzos, introdujo los mecanismos necesarios para validar campos de formulario. Estas son algunas de las validaciones típicas:

- ✓ Comprobar que se han suministrado todos los campos obligatorios.
- ✓ Comprobar que el formato de un campo es el esperado (fechas, teléfonos, etc.).
- ✓ Comprobar la validez (sintáctica) de las direcciones de correo y URL.
- ✓ Comprobar que no se sobrepasa la longitud, número de líneas o tamaño de la entrada.

12.4.1 EVENTO ONSUBMIT

La validación de campos de formulario se basa en interceptar el momento en que el usuario realiza el envío de los datos del formulario (es decir, pulsa sobre el botón Enviar). Como es sabido, el botón de envío de datos se codifica con HTML mediante un tipo especial de objeto de formulario, llamado **submit**. El siguiente ejemplo muestra un formulario con un botón de envío:

```
<form name="miFormulario" action="mailto:mi@mail.com">
<input type="submit" value="Enviar" name="enviar">
</form>
```

Cuando el usuario pulsa sobre el botón, el navegador ejecuta la acción codificada en el parámetro *action* del formulario. En el ejemplo anterior, hemos incluido una acción de tipo *mailto*., es decir, de envío de datos por correo electrónico. Lo normal es que la acción sea la invocación de un programa que se ejecuta en el servidor remoto, y que recibe los datos de acuerdo a la especificación CGI (*Common Gateway Interface*).

Cuando el usuario pulsa sobre el botón de enviar, se genera el evento **submit**, asociado al envío de datos de un formulario. JavaScript proporciona un mecanismo para capturar este evento, lo que nos permite ejecutar un *script* justo antes de que se realice el envío de los datos.

La forma de capturar el evento consiste en introducir el atributo **onSubmit** en la etiqueta del formulario cuyo evento **submit** queremos capturar. De esta forma, para capturar el evento **submit** del formulario del ejemplo anterior escribiríamos:

```
<form name="miFormulario" action="mailto:mi@mail.com"
onSubmit="alert('Has pulsado enviar.');" return false;">
<input type="submit" value="Enviar" name="enviar">
</form>
```

El código JavaScript que se ejecuta antes del envío del formulario es, en el ejemplo anterior, un *alert()*. Sin embargo, se puede observar que se ha añadido un *return false* al final de la línea de código. Esta es la forma de indicar al navegador si la validación ha sido correcta o incorrecta. Si el código JavaScript (normalmente una función de validación) del atributo *onSubmit* devuelve *false*, la validación es incorrecta y el navegador no continúa con el envío del formulario tras ejecutar dicho código. Por el contrario, si devuelve *true* la validación es correcta y el formulario se envía normalmente.

Podemos modificar el ejemplo anterior para devolver *true* tras mostrar el *alert*. El ejemplo siguiente permite comprobar lo que sucede en ese caso. Se abrirá el cliente de correo por defecto para iniciar el envío de un mensaje de correo a la dirección suministrada como *mailto*: en el atributo *action*.

```
<form name="miFormulario" action="mailto:mi@mail.com"
onSubmit="alert('Has pulsado enviar.');" return true;">
<input type="submit" value="Enviar" name="enviar">
</form>
```

12.4.2 EL OBJETO THIS

Normalmente, el evento **onSubmit** ejecuta una función de validación, como en el siguiente ejemplo:

```
form name="miFormulario" action="mailto:mi@mail.com"
onSubmit="return Valida(this);">
```



```

Campo1 <input type="text" name="campo1">
<br>
Campo2 <input type="text" name="campo2">
<input type="submit" value="Enviar" name="enviar">
</form>

```

Esta función devolverá, como ya se ha dicho, **true** si la validación es correcta, y **false** en caso contrario. Para poder realizar la validación, la función debe trabajar sobre los objetos del formulario a validar, para acceder a su valor y evaluarlo. Esto se consigue pasándole como parámetro el objeto **this**, que referencia en este contexto al formulario desde el que se invoca la función.

La función de validación accederá a los campos del formulario a partir del objeto **this**, en caso de que algunos de los campos esté vacío nos mostrará un mensaje por pantalla advirtiéndolo que falta información, en caso contrario enviará el formulario. Para ello utilizamos la instrucción *if*. La sintaxis de dicha instrucción es:

```

if (condición)
{ Sentencias si se cumple la condicion; }
else { sentencias si no cumple la condicion; }

```

Nuestra función valida quedará de la siguiente forma:

```

function Valida(formulario) {
  if ((formulario.campo1.value.length == 0) || (formulario.campo2.value.length ==0)) {
    alert('falta informacion');
    return false
  }
  return true
}
</script>

```



Esta función debe ir en la cabecera de la web, es decir entre `<head>` y `</head>`. Además, como es código JavaScript irá encerrado entre `<script>` y `</script>`.

12.4.3 VALIDACIÓN DE CAMPOS VACÍOS

Si juntamos todo lo anterior en un fichero, nos quedará el ejemplo completo de validación de campos vacíos, es decir, no podemos ningún campo de los que elijamos en blanco.

```

<html>
<head>
<script>
function Valida(formulario) {
  if ((formulario.campo1.value.length == 0) || (formulario.campo2.value.length ==0)) {
    alert('falta informacion');
    return false
  }
}

```

```

    return true
  }
</script>
<body>
<form name="miFormulario" action="mailto:mi@mail.com"
onSubmit="return Valida(this);">
Campo1 <input type="text" name="campo1">
<br>
Campo2 <input type="text" name="campo2">
<br>
<input type="submit" value="Enviar" name="enviar">
</form>
</body>
</html>

```

ACTIVIDADES 12.2



- Hacer un formulario web con 4 campos: *Nombre*, *Apellidos*, *DNI* y *Telefono*. No permitir que pueda enviar la información si alguno de los campos ha quedado en blanco.

12.4.4 VALIDACIÓN DE CAMPOS NUMÉRICOS

Para este caso nos vamos a servir de dos funciones del lenguaje JavaScript, por un lado **parseInt** y, por otro, **isNaN**. El uso de la primera viene determinado por el hecho de que todos los campos de un `<input type="text">` son considerados como texto, *parseInt* nos va a permitir convertir el valor a tipo entero. La segunda nos va a indicar si el valor que hemos convertido realmente es un número y no otra cosa. Podemos añadir esto a la función válida de nuestro ejemplo anterior para comprobar que el *campo2* debe ser de tipo numérico (ya que el resto quedaría exactamente igual):

```

function Valida(formulario) {
  if ((formulario.campo1.value.length == 0) || (formulario.campo2.value.length ==0)) {
    alert('falta informacion');
    return false;
  }
  if (isNaN(parseInt(formulario.campo2.value))) {
    alert('el campo2 debe ser un numero');
    return false;
  }
  return true
}

```

ACTIVIDADES 12.3



- Añadir al formulario de la Actividad 12.2 un campo más: *Edad*. Comprobar que el valor que hemos introducido en este campo es de tipo numérico.

12.4.5 VALIDACIÓN DE CAMPOS NUMÉRICOS DE LONGITUD FIJA

Una de las comprobaciones típicas sobre campos de formulario se refiere al número de caracteres introducidos (para verificar que no supera cierto valor, o que es exactamente igual a cierto número). Esta comprobación es útil, por ejemplo, para números de teléfono, códigos postales o para números de identificación (como el DNI). El ejemplo *Validación de campos de longitud fija* muestra cómo hacerlo con expresiones regulares.

```
function valida2(formulario) {
    var cp=/^[0-9]{5,5}$|^[0-9]{8,8}[A-Z]$/;
    var dni=/^[0-9]{8,8}[A-Z]$/;
    if (!(cp.test(formulario.campo1.value))) {
        alert('Contenido del campo1 no es un CODIGO POSTAL');
        return false; }
    if (!(dni.test(formulario.campo2.value))) {
        alert('Contenido del campo2 no es un DNI válido. ');
        return false; }
    return true;
}
```

ACTIVIDADES 12.4



- Sobre el formulario de la Actividad 12.3, comprobar que el valor que hemos introducido para el DNI y para el teléfono tienen el formato correcto.

12.4.6 VALIDACIÓN DE CAMPOS DE CORREO ELECTRÓNICO

Este caso es muy parecido al de campos de longitud fija variando la cadena a comparar. Cabe destacar que lo único que podemos comprobar es que la dirección de correo esté bien escrita, del tipo *nombre@domino.tipo*, no podemos asegurarnos de la que dirección exista y además sea de la persona que rellena el formulario.

```
function valida3(formulario) {
    var correo /^[^@\s]+@[^\s]+\.[^\s]+\.[^\s]+$/;
    if (!(correo.test(formulario.campo1.value))) {
        alert('Contenido del campo1 no es CORREO ELECTRONICO válido. ');
        return false; }
    return true;
}
```

ACTIVIDADES 12.5



- Sobre el formulario de la Actividad 12.4 añadir un nuevo campo *Email*. Comprobar que el valor que hemos introducido para dicho campo tiene un formato de correo electrónico válido.

12.5 COMPORTAMIENTOS DINÁMICOS. CAPTURA DE EVENTOS CON JAVASCRIPT

Con la aparición de Netscape Navigator 4.0 se introdujo un nuevo modelo de manejo de eventos mucho más potente que el que se venía usando anteriormente, dotando a las páginas html de mayor dinamismo y haciéndolas más vistosas. Se incluyeron nuevos eventos, un objeto *event* y la posibilidad de capturar y manejar eventos antes de que alcancen el objetivo hacia el que se lanzaron.

Las aplicaciones JavaScript no hacen más que manejar los eventos que el usuario va provocando. Cada uno de estos eventos, genera la creación de un objeto del tipo *event* que posee algunas características propias como el tipo de evento del que se trata o la localización del sitio donde ocurrió el evento. Cada evento se envía como un argumento de un manejador de eventos.

Normalmente, el objeto en el que se genera el evento es el encargado de manejarlo. Sin embargo, tal vez el usuario desee que el objeto *window* o el objeto *document* maneje ciertos eventos, que en circunstancias normales no les llegarían.

La base de este modelo es el objeto *event*, que contiene todas las propiedades que describen un evento en JavaScript y que es pasado a cada manejador de eventos cuando se produce algún evento. En el apartado anterior ya hemos visto una captura de eventos en el formulario al usar **onSubmit**.

Los principales eventos se muestran en la siguiente tabla:

Tabla 12.2 Eventos de JavaScript

Sintaxis	Descripción	Evento de...
onAbort	Ocurre cuando el usuario aborta la carga de una imagen (pulsando sobre un enlace o sobre el botón Stop).	<i>Image</i> .
onBlur	Ocurre cuando el usuario quita el foco de un elemento de un formulario, de una ventana o de un <i>frame</i> .	<i>Button, Checkbox, Password, Radio, Reset, Select, Submit, Text, Textarea</i> y <i>Window</i> .
onChange	Ocurre cuando un select, text o textArea pierde el foco y su valor ha sido modificado.	<i>FileUpload, Select, Text</i> y <i>Textarea</i> .
onClick	Ocurre cuando el usuario hace un clic sobre un enlace o un elemento de un formulario.	<i>Button, Checkbox, Link, Radio, Reset</i> y <i>Submit</i> .
onDbClick	Ocurre cuando el usuario hace un doble clic sobre un enlace o un elemento de un formulario.	<i>Document, Area</i> y <i>Link</i> .
onFocus	Ocurre cuando se produce un error en la carga de un documento o una imagen.	<i>Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea</i> y <i>Window</i> .

onKeyDown	Ocurre cuando el usuario pulsa una tecla.	<i>Document, Image, Link y Textarea.</i>
onKeyPress	Ocurre cuando el usuario pulsa o mantiene pulsada una tecla.	<i>Document, Image, Link y Textarea.</i>
onKeyUp	Ocurre cuando el usuario libera una tecla.	<i>Document, Image, Link y Textarea.</i>
onLoad	Ocurre cuando el navegador ha terminado de cargar la página.	<i>Image, Layer y Window.</i>
onMouseDown	Ocurre cuando el usuario pulsa un boton del ratón.	<i>Button, Document y Link.</i>
onMouseMove	Ocurre cuando el usuario mueve el ratón.	Se debe asociar explícitamente a algún objeto.
onMouseOut	Ocurre cuando el usuario mueve el ratón fuera de un objeto.	<i>Area, Layer y Link.</i>
onMouseOver	Ocurre cuando el usuario mueve el ratón dentro de un objeto.	<i>Area, Layer y Link.</i>
onMove	Ocurre cuando se mueve una ventana.	<i>Window.</i>
onReset	Ocurre cuando el usuario pulsa el botón <i>reset</i> de un formulario.	<i>Form.</i>
onResize	Ocurre cuando el usuario cambia el tamaño de la ventana.	<i>Window.</i>
onSelect	Ocurre cuando el usuario selecciona algún texto en un objeto <i>text</i> o <i>textarea</i> .	<i>Text y Textarea.</i>
onSubmit	Ocurre cuando el usuario envía un formulario.	<i>Form.</i>

Como ejemplos hemos visto en el apartado anterior el uso de *onSubmit*. A continuación se muestran unos ejemplos más de captura de eventos con JavaScript:

```
<body onLoad="alert('estas en mi pagina web')" onResize="alert('has cambiado el tamaño
de la pagina')">
Ejemplo de onclick <br>
<a href="http://www.google.es" onclick="alert('adios')">Me voy de la pagina</a>
<br><br>

```

ACTIVIDADES 12.6



- Haga una página al estilo de la del ejemplo anterior donde pruebe el uso varias de las capturas de eventos que hemos visto en la Tabla 12.2. Al menos debe utilizar 5 funciones distintas.

12.6 LIMITACIONES Y RIESGOS DE ATAQUES

JavaScript tiene algunas limitaciones, las cuales derivan de su objetivo principal, que es adicionar interactividad a las páginas web. JavaScript no puede interactuar con el servidor.

Una de sus principales desventajas es su fortaleza: JavaScript trabaja enteramente dentro del navegador del cliente y no se comunica con el servidor ni con otras máquinas. JavaScript no puede crear gráficos: JavaScript no pueden crear sus propios gráficos, solamente puede manipular gráficos hechos en otros programas.

JavaScript trabaja de forma diferente en diferentes navegadores: este es el principal problema, su comportamiento depende del navegador. Esta problemática se debe a que las empresas que han introducido JavaScript lo han hecho de forma independiente, sin considerar los problemas que esto trae al programador. Programar usando una versión simple de JavaScript es relativamente fácil, pero escribir código que funcione de forma notable para dos versiones dispares de navegadores es uno de los mayores desafíos para un desarrollador de páginas web. Microsoft intentó dar soporte a la primera versión de JavaScript en el Internet Explorer 3.0 (con el nombre de Jscript). Sin embargo, Jscript resultó poco fiable y estaba plagado de errores.

JavaScript fue diseñado de forma que se ejecutara en un entorno muy limitado que permitiera a los usuarios confiar en la ejecución de los *scripts*. De esta forma, los *scripts* de JavaScript no pueden comunicarse con recursos que no pertenezcan al mismo dominio desde el que se descargó el *script*. Los *scripts* tampoco pueden cerrar ventanas que no hayan abierto esos mismos *scripts*. Las ventanas que se crean no pueden ser demasiado pequeñas ni demasiado grandes ni colocarse fuera de la vista del usuario (aunque los detalles concretos dependen de cada navegador).

Además, los *scripts* no pueden acceder a los archivos del ordenador del usuario (ni en modo lectura ni en modo escritura) y tampoco pueden leer o modificar las preferencias del navegador.

Por último, si la ejecución de un *script* dura demasiado tiempo (por ejemplo, por un error de programación) el navegador informa al usuario de que un *script* está consumiendo demasiados recursos y le da la posibilidad de detener su ejecución.

A pesar de todo, existen alternativas para poder saltarse algunas de las limitaciones anteriores. La alternativa más utilizada y conocida consiste en firmar digitalmente el *script* y solicitar al usuario el permiso para realizar esas acciones.



RESUMEN DEL CAPÍTULO

En el capítulo se han visto las diferencias entre lenguajes del lado del servidor y lenguajes del lado del cliente.

De los lenguajes del lado del cliente, junto con HTML y CSS (estudiados en el capítulo anterior), uno de los más importantes y conocidos es JavaScript. En este capítulo se ha tratado JavaScript casi en exclusivo para para la resolución de problemas concretos, como son la validación de datos en los formularios y la captura de eventos. JavaScript es mucho más que eso, pero bueno, para el estudio de dicho lenguaje necesitaríamos varios libros.

Al final del capítulo hemos tratado las limitaciones y los riesgos de utilizar JavaScript. Las limitaciones le vienen sobre todo por ser un lenguaje para uso en el cliente y el hecho de no trabajar igual en todos los navegadores web.



EJERCICIOS PROPUESTOS

1. ¿Por qué cree que se usa tanto hoy en día JavaScript?
2. ¿A qué lenguaje de programación web complementa y amplía sus posibilidades?
3. ¿Qué ventajas tiene JavaScript con respecto a Java?
4. Busque alguna página en Internet con recursos en JavaScript para utilizarlos en su web.
5. Algunos navegadores, por defecto, bloquean el código JavaScript, ¿a qué puede deberse? ¿Cómo se habilita?



TEST DE CONOCIMIENTOS

1 JavaScript fue creado por:

- a) Google.
- b) Microsoft.
- c) Mozilla.
- d) Opera.
- e) Netscape.
- f) Todas son falsas.

2 El código JavaScript va incluido dentro de:

- a) `<noscript>` y `</noscript>`.
- b) `<javascript>` y `</javascript>`.
- c) `<script>` y `</script>`.
- d) Todas son ciertas.
- e) Todas son falsas.

3 Una de las principales características que nos aporta JavaScript que no soporta HTML es:

- a) Mayor profundidad de colorido.
- b) Trabajar con imágenes.
- c) Trabajar con tablas.
- d) Permite validar información de los formularios.
- e) Todas son ciertas.
- f) Todas son falsas.

4 Para convertir un dato a tipo entero necesito utilizar la función:

- a) *Alert.*
- b) *Prompt.*
- c) *Var.*
- d) *parseInt.*
- e) *Write.*
- f) *Read.*
- g) Todas son falsas.

5 Unos de los aspectos negativos de JavaScript es:

- a) Es un lenguaje orienta al servidor, por tanto necesito un servidor muy potente para poder ejecutarlo.
- b) Tiene problemas al generar imágenes.
- c) Su comportamiento es distinto según el navegador que lo interprete.
- d) Todas son ciertas.
- e) Todas son falsas.

13

Programación de documentos web utilizando lenguajes de *script* de servidor: PHP

OBJETIVOS DEL CAPÍTULO

- ✓ Conocer la clasificación de los lenguajes de programación.
- ✓ Presentar algunas de las herramientas para la edición de código.
- ✓ Estudiar el lenguaje de programación web orientado al servidor más conocido y utilizado en la actualidad: PHP.
- ✓ Realizar ejemplos y prácticas sobre dicho lenguaje.

13.1 CLASIFICACIÓN

Un lenguaje de programación es un lenguaje inventado para controlar una máquina, (normalmente, un ordenador). Hay muchísimos, de toda clase de tipos y características, inventados para facilitar el abordaje de distintos problemas, el mantenimiento del software, su reutilización, mejorar la productividad, etc.

Los lenguajes de programación se pueden clasificar según varios criterios: nivel de abstracción, propósito, evolución histórica, manera de ejecutarse, manera de abordar la tarea a realizar, paradigma de programación, lugar de ejecución, concurrencia, interactividad, realización visual, determinismo y productividad.

Hay que tener en cuenta también, que en la práctica, la mayoría de lenguajes no pueden ser puramente clasificados en una categoría, pues surgen incorporando ideas de otros lenguajes y de otras filosofías de programación, pero no importa al establecer las clasificaciones, pues el auténtico objetivo de las mismas es mostrar los rangos, las posibilidades y tipos de lenguajes que hay.

13.2 HERRAMIENTAS DE EDICIÓN DE CÓDIGO

El código PHP puede ser generado con cualquier editor de texto, es muy recomendable no utilizar editores de texto avanzados (Microsoft Word, OpenOffice Writer, etc.) ya que añaden al texto caracteres especiales que no serían “comprensibles” por el navegador. Por lo tanto, si vamos a utilizar editores de textos no específicos sería recomendable utilizar el Bloc de Notas de Windows o el Gnome Editor o Kde Editor de Linux. Existen también herramientas creadas expresamente para diseñar código PHP, hay algunas que son gratuitas y otras de pago, que se van a presentar a continuación de manera breve para que el alumno decida la que se adapta a su gusto y pueda profundizar en el manejo de dicha herramienta.

13.2.1 NOTEPAD ++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Solo funciona en Microsoft Windows. Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores. Se distribuye bajo los términos de la Licencia Pública General de GNU. Tiene multitud de lenguajes de programación soportados, incluyendo claro está PHP. Además, permite al usuario definir su propio lenguaje: no solo las palabras clave para la sintaxis coloreada, sino también las palabras clave para la envoltura de sintaxis, los comentarios clave y los operadores.



Figura 13.1. Notepad++

ACTIVIDADES 13.1



- Descargue e instale Notepad++ y haga capturas de todo el proceso, tanto en la descarga como en la instalación.

13.2.2 DREAMWEAVER

Dreamweaver es un software fácil de usar que permite crear páginas web profesionales. Las funciones de edición visual de Dreamweaver permiten agregar rápidamente diseño y funcionalidad a las páginas, sin la necesidad de programar manualmente el código HTML. Se pueden crear tablas, editar marcos, trabajar con capas, insertar comportamientos JavaScript, etc., de una forma muy sencilla y visual. Además, incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios web, actualizando el sitio web en el servidor sin salir del programa.



Figura 13.2. Dreamweaver

ACTIVIDADES 13.2



- Descargue e instale Dreamweaver y haga capturas de todo el proceso, tanto en la descarga como en la instalación.

13.2.3 ECLIPSE

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama “Aplicaciones de Cliente Enriquecido”, opuesto a las aplicaciones “Cliente-liviano” basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ), que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.



Figura 13.3. Eclipse

ACTIVIDADES 13.3



➤ Descargue e instale Eclipse haga capturas de todo el proceso, tanto en la descarga como en la instalación.

13.2.4 APTANA STUDIO

Aptana Studio es un entorno de desarrollo integrado gratuito basado en eclipse y desarrollado por Aptana, Inc., que puede funcionar bajo Windows, Mac y Linux y provee soporte para lenguajes como: PHP, Python, Ruby, CSS, Ajax, HTML y Adobe AIR. Tiene la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades. Aptana Studio es un IDE de software libre que posee la GNU General Public License o la Aptana Public License.

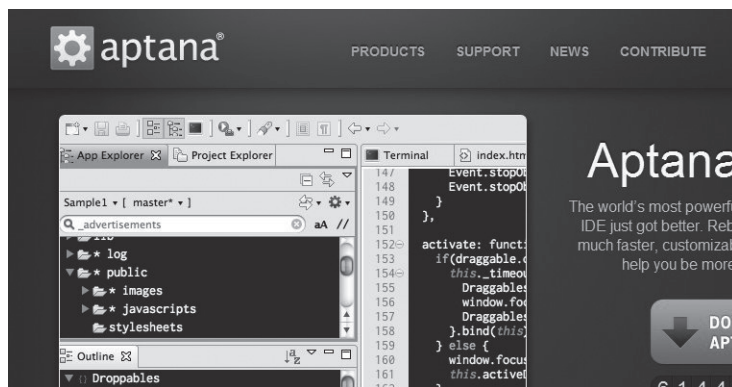


Figura 13.4. Aptana Studio

ACTIVIDADES 13.4



➤ Descargue e instale Aptana Studio y haga capturas de todo el proceso, tanto en la descarga como en la instalación.

13.3 PHP

PHP, acrónimo de *PHP Hypertext Preprocessor*, es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

Lo que distingue a PHP de la tecnología JavaScript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un *script* similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP.

13.3.1 ¿QUÉ SE PUEDE HACER CON PHP?

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un *script* CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir *cookies*.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea simple con PHP. PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. También se pueden abrir *sockets* de red directos (*raw sockets*) e interactuar con otros protocolos.

13.3.2 HISTORIA DE PHP

PHP fue concebido en otoño de 1994 por Rasmus Lerdorf. Las primeras versiones no distribuidas al público fueron usadas en sus páginas web para mantener un control sobre quién consultaba su currículum. La primera versión disponible para el público a principios de 1995 fue conocida como “Herramientas para páginas web personales” (*Personal Home Page Tools*). Consistían en un analizador sintáctico muy simple que solo entendía unas cuantas macros y una serie de utilidades comunes en las páginas web de entonces, un libro de visitas, un contador y otras pequeñas cosas. El analizador sintáctico fue reescrito a mediados de 1995 y fue nombrado PHP/FI versión 2. FI viene de otro programa que Rasmus había escrito y que procesaba los datos de formularios. Así que combinó las “Herramientas para páginas web personales”, el “intérprete de formularios”, añadió soporte para mSQL y PHP/FI vio la luz. PHP/FI creció a gran velocidad y la gente empezó a contribuir en el código.

Es difícil dar estadísticas exactas, pero se estima que a finales de 1996 PHP/FI se estaba usando al menos en 15.000 páginas web alrededor del mundo. A mediados de 1997 este número había crecido a más de 50.000. A mediados de 1997 el desarrollo del proyecto sufrió un profundo cambio, dejó de ser un proyecto personal de Rasmus, al cual habían ayudado un grupo de usuarios y se convirtió en un proyecto de grupo mucho más organizado. El analizador sintáctico se reescribió desde el principio por Zeev Suraski y Andi Gutmans y este nuevo analizador estableció las bases para PHP versión 3. Gran cantidad de código de PHP/FI fue portado a PHP3 y otra gran cantidad fue escrito completamente de nuevo.

A finales 1999, tanto PHP/FI como PHP3 se distribuyen en un gran número de productos comerciales, tales como el servidor web C2's StrongHold y Redhat Linux. En el 2007 había 20.917.850 dominios con PHP, y 1.224.183 de direcciones IP con PHP.

13.4 INTEGRACIÓN CON LOS LENGUAJES DE MARCAS. SINTAXIS

Todo código PHP siempre va “encerrado” o delimitado por los caracteres de inicio de código PHP `<?php` o simplemente `<?` y por los caracteres de fin de código PHP `?>`. Podemos utilizar PHP en solitario o junto con código HTML. Ejemplo de código PHP en solitario:

```
<?
echo "Hola, Soy un script PHP!";
?>
```



La instrucción *echo* de PHP muestra por pantalla lo que viene a continuación, en este capítulo veremos más posibilidades de funcionamiento de esta instrucción. Ejemplo: código PHP junto a código HTML.

```
<html>
<head>
<title>EJEMPLO</title>
</head>
<body>
</html>
<?php
echo "Hola, Soy un script PHP!";
?>
<P>Una parte en HTML
<?
echo "esta es una segunda parte en PHP";
?>
</body>
</html>
```

Hola, Soy un script PHP!

Una parte en HTML esta es una segunda parte en PHP

Figura 13.5. PHP

ACTIVIDADES 13.5

- Elabore también una página que mezcle PHP con HTML, la página debe mostrar por pantalla lo siguiente:
- “Esta es la primera práctica escrita en PHP” en código PHP.
 - “Escrita por Apellidos y Nombre del alumno/a” en código HTML.

13.5 COMENTARIOS

- ✓ Para realizar comentarios de una sola línea utilizaremos // ó #.
- ✓ Los comentarios de varias líneas cominezan con /* y finalizan al primer */ que se encuentre. No es conveniente abusar de este tipo de comentarios.

```
<?
echo 'Esto es una prueba';
// Esto es un comentario estilo c++ de una sola línea
/* Esto es un comentario multilínea
Si! otra línea de comentarios */
echo 'Esto es otro test';
echo 'Un test final';
# Comentario estilo consola de una sola línea
?>
```

ACTIVIDADES 13.6

- Sobre el código PHP del apartado 13.4 añada dos comentarios:
- El primero justo al principio del código PHP (después de <?), indicando que es el principio del código PHP.
 - El segundo justo antes de finalizar el código PHP (antes de ?>), indicando que es el final del código PHP.

13.6 ELEMENTOS DEL LENGUAJE ESTRUCTURADO: TIPOS DE DATOS, VARIABLES, OPERADORES, ESTRUCTURAS DE CONTROL, SUBPROGRAMAS...

13.6.1 TIPOS DE DATOS

PHP soporta los siguientes tipos: entero, números en coma flotante, cadenas, *arrays* y objetos. El tipo de una variable normalmente no lo indica el programador; en su lugar, lo decide PHP en tiempo de ejecución dependiendo del contexto en el que se utilice esa variable.

Nótese que una variable se puede comportar de formas diferentes en ciertas situaciones, dependiendo de qué tipo sea en ese momento.

ENTEROS. Los enteros se pueden especificar usando una de las siguientes sintaxis:

```
$a = 1234; # número decimal
$a = -123; # un número negativo
$a = 0123; # número octal (equivalente al 83 decimal)
$a = 0x12; # número hexadecimal (equivalente al 18 decimal)
```

NÚMEROS EN PUNTO FLOTANTE. Los números en punto flotante (*double*) se pueden especificar utilizando cualquiera de las siguientes sintaxis:

```
$a = 1.234;
$a = 1.2e3;
```

CADENAS. Las cadenas de caracteres se pueden especificar usando uno de dos tipos de delimitadores. Si la cadena está encerrada entre dobles comillas (“”), las variables que estén dentro de la cadena serán expandidas (sujetas a ciertas limitaciones de interpretación). El carácter de barra invertida (“\”) se puede usar para especificar caracteres especiales:

Tabla 13.1 Caracteres especiales

Secuencia	Significado
\n	Nueva línea.
\r	Retorno de carro.
\t	Tabulación.
\\	Barra invertida.
\s	Signo del dólar.
\"	Comillas dobles.

La segunda forma de delimitar una cadena de caracteres usa el carácter de comilla simple (‘’). Cuando una cadena va encerrada entre comillas simples, los únicos caracteres de escape que serán comprendidos son “\” y “\’”. Esto es por convenio, así que se pueden tener comillas simples y barras invertidas en una cadena entre comillas simples. Las variables no se expandirán dentro de una cadena entre comillas simples.

Se puede acceder a los caracteres dentro de una cadena tratándola como un *array* de caracteres indexado numéricamente.

ARRAYS. Los *arrays* actualmente actúan tanto como tablas *hash* (*arrays* asociativos) como *arrays* indexados (vectores). PHP soporta tanto *arrays* escalares como asociativos. De hecho, no hay diferencias entre los dos. Se puede crear una *array* usando las funciones *list()* o *array()*, o se puede asignar el valor de cada elemento del *array* de manera explícita.

```
$a[0] = "abc";
$a[1] = "def";
$b["foo"] = 13;
```

También se puede crear un *array* simplemente añadiendo valores al *array*. Cuando se asigna un valor a una variable *array* usando corchetes vacíos, el valor se añadirá al final del *array*.

```
$a[] = "hola"; // $a[2] == "hola"
$a[] = "mundo"; // $a[3] == "mundo"
```

Los *arrays* se pueden ordenar usando las funciones *asort()*, *arsort()*, *ksort()*, *rsort()*, *sort()*, *uasort()*, *usort()* y *uksort()*, dependiendo del tipo de ordenación que se desee. Se puede contar el número de elementos de un *array* usando la función *count()*. Se puede recorrer un *array* usando las funciones *next()* y *prev()*. Otra forma habitual de recorrer un *array* es usando la función *each()*.

Los *arrays* multidimensionales son bastante simples. Para cada dimensión del *array* se puede añadir otro valor [clave] al final:

```
$a[1] = $f; # ejemplos de una sola dimensión
$a["foo"] = $f;
$a[1][0] = $f; # bidimensional
$a["foo"][2] = $f;
$a[3]["bar"] = $f;
$a["foo"][4]["bar"][0] = $f; # tetradimensional!
```

13.6.2 VARIABLES

En PHP las variables se representan como un signo de dólar seguido por el nombre de la variable. El nombre de la variable es sensible a minúsculas y mayúsculas.

```
$var = "Bob";
$Var = "Joe";
echo "$var, $Var"; // produce la salida "Bob, Joe"
```

PHP ofrece otra forma de asignar valores a las variables: asignar por referencia. Esto significa que la nueva variable simplemente referencia (en otras palabras, “se convierte en un alias de” o “apunta a”) la variable original. Los cambios a la nueva variable afectan a la original, y viceversa. Esto también significa que no se produce una copia de valores; por tanto, la asignación ocurre más rápidamente. De cualquier forma, cualquier incremento de velocidad se notará solo en los bucles críticos cuando se asignen grandes *arrays* u objetos. Para asignar por referencia, simplemente se antepone un ampersand (&) al comienzo de la variable cuyo valor se está asignando (la variable fuente). Por ejemplo, el siguiente trozo de código produce la salida ‘Mi nombre es Bob’ dos veces.

13.6.3 CONSTANTES

Una constante es un identificador para expresar un valor simple. Como el nombre sugiere, este valor no puede variar durante la ejecución del *script*. Una constante es sensible a mayúsculas por defecto.

Por convención, los identificadores de constantes suelen declararse en mayúsculas. El nombre de una constante sigue las mismas reglas que cualquier etiqueta en PHP. Un nombre de constante válido empieza con una letra o un carácter de subrayado, seguido por cualquier número de letras, números, o subrayados. Se puede definir una constante usando la función *define()*. Una vez definida, no puede ser modificada ni eliminada.

Solo se puede definir como constantes valores escalares *boolean*, *integer*, *float* y *string*.

Para obtener el valor de una constante solo es necesario especificar su nombre. A diferencia de las variables, no se tiene que especificar el prefijo \$. También se puede utilizar la función *constant()*, para obtener el valor de una constante, en el caso de que queramos expresarla de forma dinámica. Usa la función *get_defined_constants()* para obtener una lista de todas las constantes definidas.

Estas son las diferencias entre constantes y variables:

- ✓ Las constantes no son precedidas por un símbolo de dolar (\$).
- ✓ Las constantes solo pueden ser definidas usando la *función()* *define*, nunca por simple asignación.
- ✓ Las constantes pueden ser definidas y accedidas sin tener en cuenta las reglas de alcance del ámbito.
- ✓ Las constantes no pueden ser redefinidas o eliminadas después de establecerse.
- ✓ Las constantes solo puede albergar valores escalares.

13.6.4 EXPRESIONES

Las expresiones son la piedra angular de PHP. En PHP, casi cualquier cosa que escribes es una expresión. La forma más simple y ajustada de definir una expresión es “cualquier cosa que tiene un valor”.

Las formas más básicas de expresiones son las constantes y las variables. Cuando escribes “\$a = 5”, estás asignando ‘5’ a \$a. ‘5’, obviamente, tiene el valor 5 o, en otras palabras ‘5’ es una expresión con el valor 5 (en este caso, ‘5’ es una constante entera).

Después de esta asignación, esperará que el valor de \$a sea 5 también, de manera que si escribes \$b = \$a, espera que se comporte igual que si escribiese \$b = 5. En otras palabras, \$a es una expresión también con el valor 5. Si todo va bien, eso es exactamente lo que pasará.

13.7 OPERADORES ARITMÉTICOS

Tabla 13.2 Operadores Aritméticos

Operación	Sintaxis	\$a	\$b	Resultado
Suma	<code>\$a+\$b</code>	12	7,3	19,3
Diferencia	<code>\$a-\$b</code>	12	7,3	4,7
Producto	<code>\$a*\$b</code>	12	7,3	87,6
Cociente	<code>\$a/\$b</code>	12	7,3	1,64383561644
Cociente entero	<code>(int)\$a/\$b</code>	12	7,3	1
Resto división	<code>\$a%\$b</code>	12	5	2
Potencia a^b	<code>pow(\$a,\$b)</code>	2	3	8
Raiz cuadrada	<code>sqrt(\$a)</code>	9		3

```
<?
# definimos dos variables numéricas asignandoles valores
$a=23;
$b=34;
/* Hacemos la suma y escribimos directamente los resultados utilizando las instrucciones
print y echo */
Echo "La suma de $a+$b es: ".$a."+".$b."=".(($a+$b));
echo "<BR>";
# guardamos el resultado en una nueva variable
$c=$a+$b;
/* ahora presentamos el resultado utilizando esa nueva variable */
echo "La suma de $a+$b es: ".$a."+".$b."=".$c;
echo "<BR>";
/* Modificamos ahora los valores de $a y $b comprobando que el cambio no modifica lo
contenido en la variable $c */
$a=513; $b=648;
print "<br> C sigue valiendo: ".$c;
echo "<br>";
?>
```

La suma de 23+34 es: 23+34=57
La suma de 23+34 es: 23+34=57

C sigue valiendo: 57

Figura 13.6. Operadores

ACTIVIDADES 13.7



- Partiendo de 2 variables *\$primera* y *\$segunda* con valores 4 y 8, respectivamente, hacer una página PHP que calcule:
- La suma de *\$primera* y *\$segunda*.
 - El producto de *\$primera* y *\$segunda*.

ACTIVIDADES 13.8



- Partiendo de 2 variables *\$primera* y *\$segunda* con valores 10 y 2, respectivamente, hacer una página PHP que calcule:
- La diferencia de *\$primera* menos *\$segunda*.
 - La división de *\$primera* entre *\$segunda*.

ACTIVIDADES 13.9



- Partiendo de 2 variables *\$primera* y *\$segunda* con valores 4 y 8, respectivamente, hacer una página PHP que calcule:
- \$primera* elevada a *\$segunda*.
 - Raíz cuadrada de *\$primera*.
 - Raíz cuadrada de *\$segunda*.

13.8 OPERADORES DE ASIGNACIÓN

El operador básico de asignación es “=”. A primera vista podría pensar que es el operador de comparación “igual que”. Pero no, realmente significa que el operando de la izquierda toma el valor de la expresión a la derecha, (esto es, “toma el valor de”).

El valor de una expresión de asignación es el propio valor asignado. Esto es, el valor de “*\$a* = 3” es 3. Esto permite hacer cosas curiosas como:

```
$a = ($b = 4) + 5;
```

```
// Ahora $a es igual a 9, y $b vale 4.
```

13.9 OPERADORES DE COMPARACIÓN

Tabla 13.3 Operadores de comparación

Nombre	Ejemplo	Resultado
Igualdad	<code>\$a == \$b</code>	Cierto si \$a es igual a \$b
Identidad	<code>\$a === \$b</code>	Cierta si \$a es igual a \$b y son del mismo tipo
Desigualdad	<code>\$a != \$b</code>	Cierto si \$a no es igual a \$b
Mayor que	<code>\$a > \$b</code>	Cierto si \$a es estrictamente mayor que \$b
Menor que	<code>\$a < \$b</code>	Cierto si \$a es estrictamente menor que \$b
Mayor o igual que	<code>\$a >= \$b</code>	Cierto si \$a es mayor o igual que \$b
Menor o igual que	<code>\$a <= \$b</code>	Cierto si \$a es menor o igual que \$b

13.10 OPERADORES DE INCREMENTO/DECREMENTO

Tabla 13.4 Operadores de incremento-decremento

Nombre	Ejemplo	Resultado
Preincremento	<code>++\$a</code>	Incrementa en 1 \$a y después devuelve \$a
Postincremento	<code>\$a++</code>	Devuelve \$a y después incrementa \$a en 1
Predecremento	<code>--\$a</code>	Decrementa en 1 \$a y después devuelve \$a
postdecremento	<code>\$a--</code>	Devuelve \$a y después decrementa \$a en 1

13.11 OPERADORES LÓGICOS

Tabla 13.5 Operadores lógicos

Nombre	Ejemplo	Resultado
Y	<code>\$a and \$b</code> <code>\$a && \$b</code>	Cierto si \$a y \$b son ciertos
O	<code>\$a or \$b</code> <code>\$a \$b</code>	Cierto si \$a o \$b son ciertos
O exclusiva	<code>\$a xor \$b</code>	Cierto si \$a o \$b son ciertos, pero no ambos a la vez
Negación	<code>!\$a</code>	Cierto si \$a no es cierto

13.12 OPERADORES DE CADENAS

Tabla 13.6 Operadores de cadenas

Nombre	Ejemplo	Resultado
Concatenación	<code>\$a . \$b</code>	Concatena \$a y \$b
Concatenación y asignación	<code>\$a .= \$b</code>	\$a tiene el contenido de lo que tenía anteriormente más lo que tiene \$b

ACTIVIDADES 13.10



- Tenemos dos cadenas `$cadena1`, con valor "hola a todo el mundo", y `$cadena2`, con valor "mi nombre es nombre y apellidos del alumno/a". Se pide:
- `$cadena3` contendrá el valor de la concatenación de `$cadena1` y `$cadena2`, mostrar por pantalla el contenido de `$cadena3`.
 - `$cadena1` contendrá el resultado de la concatenación de sí misma con `$cadena2`, mostrar por pantalla el contenido de `$cadena1`.

13.13 NÚMEROS ALEATORIOS

PHP dispone de dos funciones capaces de generar números aleatorios. Se trata de la función *rand()* y de la función mejorada *mt_rand()*. La función de PHP *mt_rand()* genera aleatorios con un algoritmo que es de promedio 4 veces más rápido que el algoritmo que utiliza *rand()*.

La forma más simple (y más desaconsejable) de generación de un número aleatorio es ésta:

```
echo rand(5,10); # número aleatorio entre 5 y 10
echo mt_rand(1,100); # número aleatorio entre 1 y 100
```

Para mejorar la aleatoriedad de *rand* podemos utilizar la siguiente función antes de generar los números aleatorios:

```
srand(time());
mt_srand(time());
```

PRÁCTICA 13.1

➤ Vamos a hacer un generador de resultados para la lotería primitiva. El juego consiste en elegir 6 números entre 1 y 49 para acertar la Combinación Ganadora en el sorteo correspondiente. Para la realización de esta práctica nos vamos a servir de las funciones vistas en este apartado tales como *mt_srand()* y *mt_rand()*.

```
<?
# generador de números para el sorteo de la primitiva
mt_srand(time()); # mejoramos la aleatoriedad
# generamos y presentamos por pantalla los 6 números
echo mt_rand(1,49);
echo ",";
echo mt_rand(1,49);
echo ",";
echo mt_rand(1,49);
echo ",";
echo mt_rand(1,49);
echo ",";
echo mt_rand(1,49);
echo ",";
echo mt_rand(1,49);
echo "<br>";
?>
```

48,36,12,5,9,47

Figura 13.7. Números aleatorios

ACTIVIDADES 13.11



- » Se pide realizar un generador para una apuesta del juego de Euromillones. Para una apuesta hay que obtener al azar cinco números principales de una lista de 50 (del 1 al 50) y además otros dos números de una lista de nueve (del 1 al 9) llamados «estrellas».

13.14 IF

La construcción *if* es una de las más importantes características de muchos lenguajes, incluido PHP. Permite la ejecución condicional de fragmentos de código. PHP caracteriza una estructura *if* que es similar a la de C:

```
if (expr) { sentencias; }
```

Por ejemplo, este código mostraría “*a* es mayor que *b*” si *\$a* fuera mayor que *\$b*, y entonces asignaría el valor de *\$a* a *\$b*:

```
if ($a > $b) {  
    print "a es mayor que b";  
    $b = $a;  
}
```

Las sentencias *if* se pueden anidar indefinidamente dentro de otras sentencias *if*, lo cual proporciona una flexibilidad completa para ejecuciones condicionales en las diferentes partes de su programa.

13.15 ELSE

Else extiende una sentencia *if* para ejecutar una sentencia en caso de que la expresión en la sentencia *if* se evalúe como FALSE. Por ejemplo, el siguiente código mostraría que *a* es mayor que *b* si *\$a* fuera mayor que *\$b*, y a NO es mayor que *b* en cualquier otro caso:

```
if ($a > $b) {  
    print "a es mayor que b";  
}  
else {  
    print "a NO es mayor que b";  
}
```

La sentencia *else* se ejecuta solamente si la expresión *if* se evalúa como FALSE, y si hubiera alguna expresión *elseif*, solo si se evaluaron también a FALSE (ver *elseif*).

13.16 ELSEIF

Elseif, como su nombre sugiere, es una combinación de *if* y *else*. Como *else*, extiende una sentencia *if* para ejecutar una sentencia diferente en caso de que la expresión *if* original se evalúa como FALSE. No obstante, a diferencia de *else*, ejecutará esa expresión alternativa solamente si la expresión condicional *elseif* se evalúa como TRUE. Por ejemplo, el siguiente código mostraría que *a* es mayor que *b*, *a* es igual a *b* o *a* es menor que *b*:

```
if ($a > $b) {  
    print "a es mayor que b";  
}  
elseif ($a == $b) {  
    print "a es igual que b";  
}  
else {  
    print "a es mayor que b";  
}
```

Puede haber varios *elseifs* dentro de la misma sentencia *if*. La primera expresión *elseif* (si hay alguna) que se evalúe como TRUE se ejecutaría. En PHP, también se puede escribir *'else if'* (con dos palabras) y el comportamiento sería idéntico al de un *'elseif'* (una sola palabra). El significado sintáctico es ligeramente distinto (si estas familiarizado con C, es el mismo comportamiento) pero la línea básica es que ambos resultarían tener exactamente el mismo comportamiento.

ACTIVIDADES 13.12



- Hacer una página PHP que compruebe si el contenido de una variable es PAR o IMPAR (recordad que en un número par su resto de la división entre 2 es 0).

ACTIVIDADES 13.13



- Hacer una página PHP que tenga dos variables y nos indique si la suma de los números es mayor que el producto de ellos o al contrario. Ejemplo:
\$numero1=1; \$numero2=2 ----> "la suma es mayor que el producto"
\$numero1=3; \$numero2=2 ----> "el producto es mayor que la suma"

ACTIVIDADES 13.14



- Hacer una página PHP que tenga dos variables y según el valor de dichas variables realice los siguientes apartados:
 - Si el valor de la primera variable es mayor que 0 y el valor de la segunda menor que la primera, mostrará el nombre del alumno/a.
 - Si el valor de la primera variable es mayor que 0 y el valor de la segunda es mayor o igual que la primera, mostrará los apellidos del alumno/a.
 - Si el valor de la primera variable es menor que 0, mostrará el nombre y apellidos completos del alumno/a, independientemente del valor de la segunda variable.

13.17 WHILE

Los bucles *while* son los tipos de bucle más simples en PHP. Se comportan como su contrapartida en C. La forma básica de una sentencia *while* es:

```
while (expr) { sentencias; }
```

El significado de una sentencia *while* es simple. Le dice a PHP que ejecute la(s) sentencia(s) anidada(s) repetidamente, mientras la expresión *while* se evalúe como TRUE. El valor de la expresión es comprobado cada vez al principio del bucle, así que incluso si este valor cambia durante la ejecución de la(s) sentencia(s) anidada(s), la ejecución no parará hasta el fin de la iteración (cada vez que PHP ejecuta las sentencias en el bucle es una iteración). A veces, si la expresión *while* se evalúa como FALSE desde el principio de todo, la(s) sentencia(s) anidada(s) no se ejecutarán ni siquiera una vez.

En el siguiente ejemplo mostrará por pantalla los números del 0 al 10 (incluidos ambos).

```
<?
$i=0;
while($i<=10) {
    echo $i;
    echo "<br>";
    /* hacemos un "intro" entre número y número*/
    $i++;
}
?>
```

0
1
2
3
4
5
6
7
8
9
10

Figura 13.8. While

ACTIVIDADES 13.15



➤ Hacer un programa en PHP que muestre por pantalla los números del 1 al 100 separados por coma “,”.

ACTIVIDADES 13.16

- Hacer una página PHP que muestre por pantalla la tabla de multiplicar de una variable (desde el 1 al 10).

ACTIVIDADES 13.17

- Hacer una página PHP que muestre por pantalla los números del 100 al 0 en orden descendente, dando el salto de 2 en 2 números. Es decir, 100, 98, 96, 94, ..., 2, 0.

ACTIVIDADES 13.18

- Hacer una página PHP que muestre una figura similar a la siguiente. **NOTA:** solo está permitido el uso de un * en todo el código.

```
*
**
***
****
*****
*****
```

ACTIVIDADES 13.19

- Hacer una página PHP que muestre por pantalla todos los divisores (aquellos cuyo resto de la división es 0) de un número guardado en una variable. Ej.: 16 -> sus divisores son: 1, 2, 4, 6, 16.

ACTIVIDADES 13.20

- Un número primo es aquel que solo es divisible entre 1 y entre sí mismo. Hacer una página PHP que muestre por pantalla si un número es o no primo.

ACTIVIDADES 13.21

- Hacer una página PHP que muestre los 20 primeros términos de la sucesión de Fibonacci. La sucesión de Fibonacci se caracteriza por tener el primer elemento: 1, el segundo elemento: 1 y el resto de elementos: se forman sumando los 2 términos anteriores -> 1, 1, 2, 3, 5, ...

13.18 FOR

Los bucles *for* son los bucles más complejos en PHP. Se comportan como su contrapartida en C. La sintaxis de un bucle *for* es:

```
for (expr1; expr2; expr3) {sentencias; }
```

La primera expresión (*expr1*) se evalúa (ejecuta) incondicionalmente una vez al principio del bucle. Al comienzo de cada iteración, se evalúa *expr2*. Si se evalúa como TRUE, el bucle continúa y las sentencias anidadas se ejecutan. Si se evalúa como FALSE, la ejecución del bucle finaliza.

Al final de cada iteración, se evalúa (ejecuta) *expr3*. Cada una de las expresiones puede estar vacía. Que *expr2* esté vacía significa que el bucle debería correr indefinidamente (PHP implícitamente lo considera como TRUE, al igual que C). Esto puede que no sea tan inútil como se podría pensar, puesto que a menudo se quiere salir de un bucle usando una sentencia *break* condicional en vez de usar la condición de *for*. Considera los siguientes ejemplos. Todos ellos muestran números del 1 al 10:

```
/* ejemplo 1 */
for ($i =1; $i<=10;$i++) {
    print $i;
}

/* ejemplo 3 */
$i=1;
for (;;) {
    if ($i>10)    break;
    print $i;
    $i++;
}

/* ejemplo 2 */
for ($i=1;;$i++)
{
    if ($i>10) {
        break;
    }
    print $i;
    $i++;
}

/* ejemplo 4 */
for ($i=1;$i<=10;print$i, $i++);
```

ACTIVIDADES 13.22



➤ Hacer un programa en PHP que muestre por pantalla los números del 1 al 100 separados por coma “,”.

ACTIVIDADES 13.23

- Hacer una página PHP que muestre por pantalla la tabla de multiplicar de una variable (desde el 1 al 10).

ACTIVIDADES 13.24

- Hacer una página PHP que muestre por pantalla los números del 100 al 0 en orden descendente, dando el salto de 2 en 2 números. Es decir, 100, 98, 96, 94, ..., 2, 0.

ACTIVIDADES 13.25

- Hacer una página PHP que muestre una figura similar a la siguiente. **NOTA:** solo está permitido el uso de un * en todo el código.

```
*
**
***
****
*****
*****
```

ACTIVIDADES 13.26

- Hacer una página PHP que muestre por pantalla todos los divisores (aquellos cuyo resto de la división es 0) de un número guardado en una variable. Ej.: 16 -> sus divisores son: 1, 2, 4, 6, 16.

ACTIVIDADES 13.27

- Un número primo es aquel que solo es divisible entre 1 y entre sí mismo. Hacer una página PHP que muestre por pantalla si un número es o no primo.

ACTIVIDADES 13.28

- Hacer una página PHP que muestre los 20 primeros términos de la sucesión de Fibonacci. La sucesión de Fibonacci se caracteriza por tener el primer elemento: 1, el segundo elemento: 1 y el resto de elementos: se forman sumando los 2 términos anteriores -> 1, 1, 2, 3, 5, ...

13.19 SWITCH

La sentencia *switch* es similar a una serie de sentencias *if* en la misma expresión. En muchas ocasiones, se quiere comparar la misma variable (o expresión) con muchos valores diferentes y ejecutar una parte de código distinta dependiendo de a qué valor es igual. Para ello sirve la sentencia *switch*.

Los siguientes dos ejemplos son dos modos distintos de escribir la misma cosa, uno usa una serie de sentencias *if*, y el otro usa la sentencia *switch*:

If	Swich
<pre>if (\$i == 0) { print "i es igual a 0"; } if (\$i == 1) { print "i es igual a 1"; } if (\$i == 2) { print "i es igual a 2"; }</pre>	<pre>switch(\$i) { case 0: print "i es igual a 0"; break; case 1: print "i es igual a 1"; break; case 2: print "i es igual a 2"; break; }</pre>

Es importante entender cómo se ejecuta la sentencia *switch* para evitar errores. La sentencia *switch* ejecuta línea por línea (realmente, sentencia a sentencia). Al comienzo, no se ejecuta código. Solo cuando se encuentra una sentencia *case* con un valor que coincide con el valor de la expresión *switch* PHP comienza a ejecutar las sentencias. PHP continúa ejecutando las sentencias hasta el final del bloque *switch* o la primera vez que vea una sentencia *break*. Si no se escribe una sentencia *break* al final de una lista de sentencias *case*, PHP seguirá ejecutando las sentencias del siguiente *case*. Por ejemplo:

```
switch ($i) {  
    case 0:  
        print "1 es igual a 0";  
    case 1:  
        print "1 es igual a 1";  
    case 2:  
        print "1 es igual a 2";  
}
```

Aquí, si *\$i* es igual a 0, ¡PHP ejecutaría todas las sentencias *print!*. Si *\$i* es igual a 1, PHP ejecutaría las últimas dos sentencias *print* y solo si *\$i* es igual a 2, se obtendría la conducta “esperada” y solamente se mostraría “*i* es igual a 2”. Así, es importante no olvidar las sentencias *break* (incluso aunque pueda querer evitar escribirlas intencionadamente en ciertas circunstancias).

En una sentencia *switch*, la condición se evalúa solo una vez y el resultado se compara a cada sentencia *case*. En una sentencia *elseif*, la condición se evalúa otra vez. Si tu condición es más complicada que una comparación simple y/o está en un bucle estrecho, un *switch* puede ser más rápido. La lista de sentencias de un *case* puede también estar vacía, lo cual simplemente pasa el control a la lista de sentencias del siguiente *case*.

<pre>/* ejemplo 1*/ switch (\$i) { case 0: case 1: case 2: print "i es menor que 3, pero no negativo"; break; case 3: print "i es 3"; }</pre>	<pre>/*ejemplo 2*/ switch (\$i) { case 0: print "i es 0"; break; case 1: print "i es 1"; break; case 2: print "i es 2"; break; default: print "i no 0, 1 o 2"; }</pre>
---	--

La expresión *case* puede ser cualquier expresión que se evalúe a un tipo simple, es decir, números enteros o de punto flotante y cadenas de texto. No se pueden usar aquí ni *arrays* ni objetos a menos que se conviertan a un tipo simple.

ACTIVIDADES 13.29



➤ Hacer una página PHP que para una nota almacenada en una variable, muestre por pantalla lo siguiente:

- Si la nota es menor que 5 → “suspense”.
- Si la nota está entre 5 y 6 → “aprobado”.
- Si la nota está entre 6 y 7 → “bien”.
- Si la nota está entre 7 y 8 → “notable”.
- Si la nota es mayor que 8 → “sobresaliente”.

13.20 FUNCIONES INTEGRADAS Y DE USUARIO

Una función se define con la siguiente sintaxis:

```
function func ($arg_1, $arg_2, ..., $arg_n) {
    echo "Función de ejemplo.";
    return $retval;
}
```

Cualquier instrucción válida de PHP puede aparecer en el cuerpo de la función, incluso otras funciones y definiciones de clases. PHP no soporta la sobrecarga de funciones y tampoco es posible redefinir u ocultar funciones previamente declaradas.

13.20.1 PARÁMETROS DE LAS FUNCIONES

La información puede suministrarse a las funciones mediante la lista de parámetros, una lista de variables y/o constantes separadas por comas. PHP soporta pasar parámetros por valor (el comportamiento por defecto), por referencia y parámetros, por defecto.

13.20.2 PASAR PARÁMETROS POR REFERENCIA

Por defecto, los parámetros de una función se pasan por valor (de manera que si cambias el valor del argumento dentro de la función, no se ve modificado fuera de ella). Si deseas permitir a una función modificar sus parámetros, debes pasarlos por referencia.

Si quieres que un parámetro de una función siempre se pase por referencia, puedes anteponer un *ampersand* (&) al nombre del parámetro en la definición de la función:

```
function add_some_extra(&$string) {
    $string .= ' y algo más.';
}
$str = 'Esto es una cadena, ';
add_some_extra($str);
echo $str; // Saca 'Esto es una cadena, y algo más.'
```

Si deseas pasar una variable por referencia a una función que no toma el parámetro por referencia por defecto, puedes anteponer un *ampersand* al nombre del parámetro en la llamada a la función:

```
function foo ($bar) {
    $bar .= ' y algo más.';
}
$str = 'Esto es una cadena, ';
foo ($str);
echo $str; // Saca 'Esto es una cadena, '
foo (&$str);
echo $str; // Saca 'Esto es una cadena, y algo más.'
```

13.20.3 PARÁMETROS POR DEFECTO

Una función puede definir valores por defecto para los parámetros escalares estilo C++:

```
function makecoffee ($type = "cappucino") {  
    return "Hacer una taza de $type.\n";  
}  
echo makecoffee ();  
echo makecoffee ("espresso");
```

La salida del fragmento anterior es:

```
Hacer una taza de cappucino.  
Hacer una taza de espresso.
```

13.20.4 DEVOLVER VALORES

Los valores se retornan usando la instrucción opcional *return*. Puede devolverse cualquier tipo de valor, incluyendo listas y objetos.

```
function square ($num) {  
    return $num * $num;  
}  
echo square (4);  
// saca '16'.
```

No puedes devolver múltiples valores desde una función, pero un efecto similar se puede conseguir devolviendo una lista.

```
function small_numbers() {  
    return array (0, 1, 2);  
}  
list ($zero, $one, $two) = small_numbers();
```

A continuación un ejemplo de la función *doble*, que recibe un número y nos devuelve su doble.

```
<?  
function doble($numero)  
{  
    return ($numero*2);  
}  
$b=7;  
$c=doble($b);  
echo "el numero $b, y su doble $c";  
?>
```

ACTIVIDADES 13.30



- Hacer una página PHP que contenga una función primo que recibe un número como parámetro y devuelve los siguientes resultados:
 - a. Si el número es primo muestra por pantalla "El número es primo".
 - b. Si el número no es primo muestra por pantalla "El número no es primo".

ACTIVIDADES 13.31



- Hacer una página PHP que contenga una función media que recibe 4 números como parámetros y devuelve el valor medio de los 4 parámetros recibidos (nota la media de 4 números es la suma de los 4 dividida entre 4).

ACTIVIDADES 13.32



- Hacer una página PHP que contenga una función cuentavocales que recibe una cadena de texto y muestra por pantalla el número de vocales totales que tiene la cadena (recordad que una cadena de texto se comporta como un *array* de caracteres).

ACTIVIDADES 13.33



- Hacer una página PHP que contenga una función cuadrado que recibe 2 parámetros, un carácter (que puede ser cualquiera) y un número, la función debe mostrar por pantalla un cuadrado con el carácter recibido (tantas filas y columnas como indique el número).

Ejemplo: cuadrado("#",3)

```
###
###
###
```

ACTIVIDADES 13.34



- Hacer una página PHP que contenga una función pirámide que recibe 2 parámetros, un carácter (que puede ser cualquiera) y un número, la función debe mostrar por pantalla una pirámide invertida con el carácter recibido (tantas filas y columnas como indique el número y de ahí descendiendo hasta 1).

Ejemplo: pirámide("@",4)

```
@@@@
@@@
@@
@
@
```

ACTIVIDADES 13.35

- Hacer una página PHP que contenga una función tiempo que recibe 2 parámetros, el tiempo que hizo ayer y el tiempo que hace hoy. Hay 3 posibilidades: SOL, NUBLADO, LLUVIA. Y en función del tiempo de ayer y el tiempo de hoy muestre la predicción para mañana. Esta predicción se basa en la siguiente tabla.

AYER, HOY → MAÑANA

SOL, SOL → SOL

NUBLADO, NUBLADO → NUBLADO

LLUVIA, LLUVIA → LLUVIA

SOL, NUBLADO → NUBLADO

NUBLADO, SOL → SOL

SOL, LLUVIA → NUBLADO

LLUVIA, SOL → NUBLADO

NUBLADO, LLUVIA → LLUVIA

LLUVIA, NUBLADO → NUBLADO

13.21 INCLUDE

Este tipo de ficheros pueden contener: texto, etiquetas HTML y funciones. Si no contiene funciones se podrá insertar tantas veces como se invoque y se insertará, además, todo su contenido.

Si el fichero contiene funciones solo podrá ser invocado una vez ya que si se hiciera una segunda llamada se produciría un error por duplicidad en los nombres de las funciones.

Es posible incluir cualquier tipo de funciones. Vamos a ver en primer lugar el fichero con las funciones y, en segundo lugar, su utilización en una página PHP. Para diferenciar los ficheros a incluir del resto de páginas PHP se puede optar por añadir antes de la extensión *.php* las iniciales *.inc* (de fichero para incluir), no es obligatorio, pero sí aconsejable.

```
<?
/* Fichero funciones.inc.php */
function doble($numero) {
return ($numero*2); }
function mitad($numero) {
return (int)($numero/2); }
?>

<?
/* fichero prueba.php */
include("funciones.inc.php"); # utilizamos include al principio del fichero
$a=6;
```

```
echo $a;
echo "<br>";
echo doble($a);
echo "<br>";
echo mitad($a);
?>
```

6
12
3

Figura 13.9

Definidas las funciones en el fichero a incluir y colocado al comienzo de la página un *script* que contenga el *include* y la ruta de ese fichero, bastará con invocar cualquiera de las funciones, en cualquier punto del documento, para que esa llamada sea sustituida por el resultado de la ejecución de la función.

Como es lógico, solo serán visualizados en el navegador del cliente los resultados de la ejecución de las funciones que hayan sido invocadas.

ACTIVIDADES 13.36



➤ Crear un fichero llamado *funciones.inc.php* que incluya dentro de él las siguientes funciones:

- la función creada en las Actividades 13.30.
- la función creada en las Actividades 13.31.
- la función creada en las Actividades 13.32.
- la función creada en las Actividades 13.33.

ACTIVIDADES 13.37



➤ Crear un fichero llamado *pruebafunciones.php* que incluya dentro de él una llamada con *include()* a la actividad anterior, comprobando cada una de las funciones.

13.22 REQUIRE

Tiene la misma sintaxis que `include` y una funcionalidad similar, aunque con algunas diferencias. Igual que ocurría con aquél, cuando un fichero es invocado por `require` esa llamada lo que hace es sustituirse a sí misma por el contenido del fichero especificado.

A diferencia de *include*, la etiqueta `require` lee y escribe (en el documento a partir del que es invocada) el archivo referenciado completo y no acepta condicionales que sí son aceptados por *include*.

ACTIVIDADES 13.38



- Crear un fichero llamado *pruebafunciones.php* que incluya dentro de él una llamada con `require()` a la Actividad 13.36, comprobando cada una de las funciones.

13.23 INCLUDE_ONCE() Y REQUIRE_ONCE()

Tanto en el caso de usar la instrucción `include` como con `require`, si se intenta incluir dos o más veces un fichero que contenga funciones, se producirá un error (PHP no permite que dos funciones tengan el mismo nombre) y se interrumpirá la ejecución del *script*. Los errores de ese tipo pueden evitarse usando las funciones:

```
include_once("fichero")
```

```
require_once("fichero")
```

Que a diferencia de *incluye()* y *require()* van a impedir que un mismo fichero pueda incluirse dos veces.

ACTIVIDADES 13.39



- Crear un fichero llamado *pruebafunciones.php* que incluya dentro de él una llamada con `include_once()` a la Actividad 13.36, comprobando cada una de las funciones.

ACTIVIDADES 13.40



- Crear un fichero llamado *pruebafunciones.php* que incluya dentro de él una llamada con `require_once()` a la Actividad 13.36, comprobando cada una de las funciones.

13.24 ELEMENTOS DE ORIENTACIÓN A OBJETO

La programación orientada a objetos es una metodología de programación avanzada y bastante extendida, en la que los sistemas se modelan creando clases, que son un conjunto de datos y funcionalidades. Las clases son definiciones, a partir de las que se crean objetos. Los objetos son ejemplares de una clase determinada y, como tal, disponen de los datos y funcionalidades definidos en la clase.

La programación orientada a objetos permite concebir los programas de una manera bastante intuitiva y cercana a la realidad. La tendencia es que un mayor número de lenguajes de programación adopten la programación orientada a objetos como paradigma para modelizar los sistemas. Prueba de ello es la nueva versión de PHP (5), que implanta la programación de objetos como metodología de desarrollo. También Microsoft ha dado un vuelco hacia la programación orientada a objetos, ya que .NET dispone de varios lenguajes para programar y todos orientados a objetos.

Así pues, la programación orientada a objetos es un tema de gran interés, pues es muy utilizada y cada vez resulta más esencial para poder desarrollar en casi cualquier lenguaje moderno. En este artículo vamos ver algunas nociones sobre la programación orientada a objetos en PHP. Aunque es un tema bastante amplio, novedoso para muchos y en un principio, difícil de asimilar, vamos a tratar de explicar la sintaxis básica de PHP para utilizar objetos, sin meternos en mucha teoría de programación orientada a objetos en general.

13.24.1 LAS CLASES: CLASS

Una clase es un conjunto de variables, llamados atributos, y funciones, llamadas métodos, que trabajan sobre esas variables. Las clases son, al fin y al cabo, una definición: una especificación de propiedades y funcionalidades de elementos que van a participar en nuestros programas.

Por ejemplo, la clase *Caja* tendría como atributos características como las dimensiones, color, contenido y cosas semejantes. Las funciones o métodos que podríamos incorporar a la clase *caja* son las funcionalidades que deseamos que realice la caja, como *introduce()*, *muestra_contenido()*, *comprueba_si_cabe()*, *vaciate()*...

Las clases en PHP se definen de la siguiente manera:

```
<?
class Caja{
    var $alto;
    var $ancho;
    var $largo;
    var $contenido;
    var $color;
function introduce($cosa){
    $this->contenido = $cosa;
}
function muestra_contenido(){
    echo $this->contenido;
}
}
?>
```

En este ejemplo se ha creado la clase *Caja*, indicando como atributos el ancho, alto y largo de la caja, así como el color y el contenido. Se han creado, para empezar, un par de métodos, uno para introducir un elemento en la caja y otro para mostrar el contenido.

Si nos fijamos, los atributos se definen declarando unas variables al principio de la clase. Los métodos se definen declarando funciones dentro de la clase. La variable *\$this*, utilizada dentro de los métodos la explicaremos un poco más abajo.

ACTIVIDADES 13.41



- Complete la clase *Caja* con la función *introducecompleta*, donde no solo se introduce el contenido, sino que también se introduce el alto, ancho, largo y color.

ACTIVIDADES 13.42



- Complete la clase *Caja* con la función *muestra*, donde se muestran todos los elementos de la clase, es decir, el alto, ancho, largo contenido y color.

13.24.2 UTILIZAR LA CLASE

Las clases solamente son definiciones. Si queremos utilizar la clase tenemos que crear un ejemplar de dicha clase, lo que corrientemente se le llama instanciar un objeto de una clase.

```
$micaja = new Caja;
```

Con esto hemos creado, o mejor dicho, instanciado, un objeto de la clase *Caja* llamado *\$micaja*.

```
$micaja->introduce("algo");
$micaja->muestra_contenido();
```

Con estas dos sentencias estamos introduciendo “algo” en la caja y luego estamos mostrando ese contenido en el texto de la página. Nos fijamos que los métodos de un objeto se llaman utilizando el código “->”.

```
nombre_del_objeto->nombre_de_metodo()
```

Para acceder a los atributos de una clase también se accede con el código “->”. De esta forma:

```
nombre_del_objeto->nombre_del_atributo
```

13.24.3 LA VARIABLE \$THIS

Dentro de un método, la variable *\$this* hace referencia al objeto sobre el que invocamos el método. En la invocación *\$micaja->introduce("algo")* se está llamando al método *introduce* sobre el objeto *\$micaja*. Cuando se está ejecutando ese método, se vuelca el valor que recibe por parámetro en el atributo *contenido*. En ese caso *\$this->contenido* hace referencia al atributo *contenido* del objeto *\$micaja*, que es sobre el que se invocaba el método.

ACTIVIDADES 13.43



- Hacer una página PHP que sea un caso completo de la clase *Caja*, donde se cree un objeto de la clase *Caja*, se le introduzcan elementos y se muestren por pantalla dichos elementos.

13.25

GESTIÓN DE ERRORES

Los errores más importantes que podemos encontrarnos en PHP son:

- E_ERROR: errores fatales en tiempo de ejecución.
- E_WARNING: advertencia en tiempo de ejecución (no son fatales).
- E_PARSE: errores fatales en tiempo de compilación.
- E_NOTICE: avisos en tiempo de ejecución (menos importantes que las advertencias).

Para que entre en funcionamiento nuestro manejador tendremos que decírselo al servidor, para ello utilizaremos la función `set_error_handler()` a la cual le pasaremos como parámetro el nombre de nuestra función que hará de manejador de errores.

Si aparte de todo esto, queremos evitar que salga por pantalla cualquier error propio de php, bastará con colocar en los archivos correspondientes la instrucción:

```
error_reporting(0);
```

13.26

EXCEPCIONES EN PHP

Para una buena programación sería conveniente crearnos nuestras propias excepciones y utilizarlas durante toda la aplicación.

Una excepción es como una clase normal y dentro de ella cada uno puede añadir, borrar o modificar a su gusto.

```
try {
    if(!@include('/path/to/fichero.php')) {
        throw new Exception('Error al cargar el fichero');
    }
}
catch(Exception $e) {
    print $e->getMessage();
}
```

Con un poco de análisis podemos gestionar los errores de nuestra aplicación de forma eficiente y muy rápida.

13.27 FORMULARIOS. PROCESAMIENTO DE DATOS RECIBIDOS DESDE EL CLIENTE. MÉTODOS DE ENVÍO DE DATOS DESDE EL CLIENTE AL SERVIDOR

13.27.1 ENVÍO A TRAVÉS DEL NAVEGADOR

La forma más simple de que un cliente pueda enviar valores a un servidor es incluir esos valores en la propia petición, insertándolos directamente en la barra de direcciones del navegador, aunque de hecho, vamos a utilizar los formularios HTML para el envío de dicha información. Dentro del formulario podemos enviar los datos de 2 formas distintas: Utilizando en método GET o utilizando en método POST. Esto se lo indicaremos en el formulario de la siguiente forma:

```
<FORM NAME="formulario" METHOD="GET" ACTION="pagina.php">
<FORM NAME="formulario" METHOD="POST" ACTION="pagina.php">
```

13.27.2 MÉTODO GET

No se diferencia en nada del descrito para el supuesto anterior. Utiliza las mismas variables predefinidas, las utiliza con idéntica sintaxis y se comporta de igual forma en lo relativo a las opciones de *register_globals*. Los nombres de las variables son en este caso, los incluidos como *name* en cada una de las etiquetas del formulario.

Respecto a los valores de cada variable, estos serían los recogidos del formulario. En los casos de campos tipo: *text*, *password* y *textarea* serían los valores introducidos por el usuario en cada uno de esos campos. En el caso de los campos tipo *radio* (en el que varias opciones pueden tener el mismo nombre) recogería el valor indicado en la casilla marcada; mientras que si se trata de campos tipo *checkbox* se transferirían únicamente las variables (y los valores) que corresponden a las casillas marcadas. Si se tratara de un campo tipo *hidden* se transferiría el valor contenido en su etiqueta y, por último, en el caso del *select* sería transferido como valor de la variable la parte del formulario contenida entre las etiquetas `<option></option>` de la opción seleccionada.

13.27.3 MÉTODO POST

En el caso de que el método de envío sea POST hay una diferencia a tener en cuenta en cuanto a las variables que recogen la información. Ahora será:

```
$HTTP_POST_VARS['n1']
```

Y ocurrirá algo similar con las superglobales, que pasarían a ser del tipo:

```
$_POST['n1'].
```

Si *register_globals* está en *On* el comportamiento de las variables directas es idéntico con ambos métodos.

13.27.4 IDENTIFICACIÓN DEL MÉTODO DE ENVÍO

PHP recoge en una variable el método utilizado para enviar los datos desde un formulario. Se trata de la variable `REQUEST_METHOD`. Puede ser invocada como una variable directa (en caso de que *register globals* esté en *on*) o a través de una de las variables de servidor.

En el primer caso la variable se llamaría: `$REQUEST_METHOD`

y en el segundo: `$HTTP_SERVER_VARS[REQUEST_METHOD]`

Cuando PHP permita el uso de variables superglobales se puede utilizar: `$_SERVER[REQUEST_METHOD]`.

Una advertencia importante. Observe que en este caso no se incluyen comillas dentro del corchete, como ocurría con todos los nombres de variable anteriores.

13.27.5 DIFERENCIAS ENTE LOS MÉTODOS GET Y POST

MÉTODO GET. Al ser enviado el formulario se carga en el navegador la dirección especificada como *action*, se le añade una interrogación (?) y, a continuación, se incluyen los datos del formulario. Todos los datos de la petición van a ser visibles desde la barra de direcciones del navegador. Únicamente son aceptados los caracteres ASCII.

Tiene una limitación en el tamaño máximo de la cadena que contiene los datos a transferir. En IE esa limitación es de 2.083 caracteres.

MÉTODO POST. No tiene las limitaciones indicadas para el caso de GET en lo relativo a visibilidad ni en cuanto a aceptación de caracteres no ASCII. Este método de transferencia de datos es el más habitual cuando se utilizan formularios.

13.27.6 LA VARIABLE \$_REQUEST

PHP también dispone (a partir de su versión 4.1.0) de la variable `$_REQUEST` (de tipo superglobal) que aúna las funcionalidades de `$_GET` y `$_POST` y que recoge en variables del tipo `$_REQUEST['nombre']` tanto los valores transferidos mediante el método GET como mediante POST.

`$_REQUEST`, a diferencia de `$_GET` y `$_POST`, no dispone de equivalentes en versiones anteriores. Ello quiere decir, no existe una variable (no superglobal) del tipo `$HTTP_REQUEST_VARS` con `$HTTP_GET_VARS` o con `$HTTP_POST_VARS`.

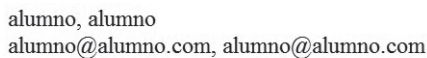
Para el siguiente ejemplo vamos a utilizar 2 páginas, la primera es una página HTML, que contiene un formulario, en el que utilizando en método de envío POST, enviaremos los datos otra página, esta PHP, que recibe los datos y los muestra por pantalla. La página PHP recibe los datos utilizando `$_POST` y `$_REQUEST`.

```
<form name="datos" method="POST" action="formu2.php">
nombre
<input type="text" name="nombre">
<br>
email
<input type="text" name="email">
<br>
<input type="submit" name="enviar" value="enviar">
</form>
```

```
<?
/* fichero formu2.php */
$nombre=$_POST['nombre'];
$email=$_POST['email'];
$nombre2=$_REQUEST['nombre'];
$email2=$_REQUEST['email'];
echo "$nombre, $nombre2";
echo "<br>";
echo "$email, $email2";
?>
```



Figura 13.10. Formulario



```
alumno, alumno
alumno@alumno.com, alumno@alumno.com
```

Figura 13.11. formu2.php

ACTIVIDADES 13.44



- Hacer un formulario PHP que pida una serie de datos: nombre, apellidos, dirección, teléfono, población y provincia. Este formulario enviará los datos a otra página PHP mediante el método GET. La página que recibe los datos debe mostrar los datos recibidos por pantalla.

ACTIVIDADES 13.45



- Hacer un formulario PHP que pida una serie de datos: nombre, apellidos, dirección, teléfono, población y provincia. Este formulario enviará los datos a otra página PHP mediante el método POST. La página que recibe los datos debe mostrar los datos recibidos por pantalla.

13.28 USO DE LA FUNCIÓN ISSET

Hasta el momento nos hemos servido de 2 ficheros para llevar a cabo nuestras páginas, uno era un formulario que recogía la información y, el otro, una página PHP que procesaba dicha información.

A partir de ahora nos vamos a servir de la función **isset** de PHP que nos va a permitir integrar esos 2 ficheros en uno solo. La función **isset** significa tiene asignado valor y la podremos utilizar para testear si hemos presionado o no el botón enviar del formulario “submit”. Si el botón “submit” tiene valor (*isset*) quiere decir que hemos pulsado sobre él, en caso contrario es que no hemos pulsado sobre dicho botón. Podemos ver el caso práctico anterior resuelto en un solo fichero.

```
<?
if (isset($_POST['enviar']))
{
/* estamos aquí porque hemos pulsado el boton enviar */
$nombre=$_POST['nombre'];
$email=$_POST['email'];
echo "$nombre";
echo "<br>";

echo "$email";
}
else
{
/* si estamos aquí es porque enviar no ha sido pulsado */
?>
<form name="datos" method="POST" action="">
nombre
<input type="text" name="nombre">
<br>
email
<input type="text" name="email">
<br>
<input type="submit" name="enviar" value="enviar">
</form>
<?
}
?>
```



A screenshot of a web form. It has two text input fields. The first is labeled 'nombre' and contains the text 'alumno'. The second is labeled 'email' and contains the text 'alumno@alumno.com'. Below these fields is a button labeled 'enviar'.

Figura 13.12. isset



A screenshot of a web page showing the output of the form. It displays the text 'alumno' on the first line and 'alumno@alumno.com' on the second line.

Figura 13.13. isset



Se puede observar como ahora el **action** de nuestra página es `""`. Esto indica que nos enviamos la información a nosotros mismos. También podríamos haber utilizado el nombre de nuestra página dentro del *action*, pero de esta forma es más cómodo, ya que si cambiamos el nombre a nuestra página, la redirección seguirá funcionando.

13.29 SESIONES

Las sesiones, en aplicaciones web realizadas con PHP y en el desarrollo de páginas web en general, nos sirven para almacenar información que se memorizará durante toda la visita de un usuario a una página web. Dicho de otra forma, un usuario puede ver varias páginas durante su paso por un sitio web y con sesiones podemos almacenar variables que podremos acceder en cualquiera de esas páginas.

Digamos que las sesiones son una manera de guardar información, específica para cada usuario, durante toda su visita. Cada usuario que entra en un sitio abre una sesión, que es independiente de la sesión de otros usuarios. En la sesión de un usuario podemos almacenar todo tipo de datos, como su nombre, productos de un hipotético carrito de la compra, preferencias de visualización o trabajo, páginas por las que ha pasado, etc. Todas estas informaciones se guardan en lo que denominamos variables de sesión.

PHP dispone de un método bastante cómodo de guardar datos en variables de sesión y de un juego de funciones para el trabajo con sesiones y variables de sesión. Lo veremos en este artículo.

13.29.1 TRABAJO CON SESIONES EN PHP

Cuando queremos utilizar variables de sesión en una página tenemos que iniciar la sesión con la siguiente función:

```
session_start ( );
```

Inicia una sesión para el usuario o continúa la sesión que pudiera tener abierta en otras páginas. Al hacer *session_start()* PHP internamente recibe el identificador de sesión almacenado en la *cookie* o el que se envíe a través de la URL. Si no existe tal identificador se sesión, simplemente lo crea.



Si en el **php.ini** se ha definido la variable *session.auto_start* = 1 se inicializa automáticamente la sesión en cada página que visita un usuario, sin que se tenga que hacer el *session_start()*.

Una vez inicializada la sesión con `session_start()` podemos a partir de ahora utilizar variables de sesión, es decir, almacenar datos para ese usuario, que se conserven durante toda su visita o recuperar datos almacenados en páginas que haya podido visitar.

La sesión se tiene que inicializar antes de escribir cualquier texto en la página. Esto es importante y de no hacerlo así corremos el riesgo de recibir un error, porque al iniciar la sesión se deben leer las *cookies* del usuario, algo que no se puede hacer si ya se han enviado las cabeceras del HTTP.



Si se intenta abrir una sesión después de haber enviado texto de la página al cliente se obtendrá el siguiente mensaje:
Warning: session_start(): Cannot send session cache limiter - headers already sent (output started at ...)

Una vez iniciada la sesión podemos utilizar variables de sesión a través de `$_SESSION`, que es un *array* asociativo, donde se accede a cada variable a partir de su nombre, de este modo:

```
$_SESSION["nombre_de_variable"]
```



`$_SESSION` es una variable global que existe a partir de PHP 4.1.0. Lo normal es que podamos acceder a esa variable normalmente, pero si nuestra versión de PHP no está actualizada podemos intentarlo con `$HTTP_SESSION_VARS`, que es también un *array* asociativo, aunque no es de ámbito global.

Si `$HTTP_SESSION_VARS` tampoco funciona tendremos que registrar cada variable por separado con la función `session_register()`, enviando por parámetro los nombres de las variables de sesión que desea utilizar desde PHP.

Ejemplo de código para definir una variable de sesión:

```
<?
session_start();
?>
<html>
<head>
<title>Generar variable de sesión</title>
</head>
<body>
<?
$_SESSION["mivariabledesesion"] = "Hola este es el valor de la variable de sesión";
?>
</body>
</html>
```

Como se puede ver, es importante inicializar la sesión antes de hacer otra cosa en la página. Luego podremos definir variables de sesión en cualquier lugar del código PHP de la página.

Para leer una variable de sesión se hace a través del mismo *array* asociativo `$_SESSION`. Es tan sencillo como haríamos para utilizar cualquier otra variable, lo único es que tenemos que haber inicializado la sesión previamente. Y, por supuesto, que la variable que deseamos acceder exista previamente.

```
<?
session_start();
?>
<html>
<head>
<title>Leo variable de sesión</title>
</head>
<body>
Muestro esa variable:
<?
echo $_SESSION["mivariabledesesion"];
?>
</body>
</html>
```



Si intentamos acceder a una variable de sesión con `$_SESSION` que no ha sido creada obtendremos otro mensaje de error:

```
Notice: Undefined index: mivariabledesesion,
```

Que es el mismo que si intentamos acceder a cualquier elemento de un *array* que no existe.

Otras funciones útiles para la gestión de sesiones son:

Tabla 13.7 Funciones útiles gestión de sesiones

Función	Descripción
Session_id()	Nos devuelve el identificador de la sesión.
Session_destroy()	Dé por abandonada la sesión eliminando variables e identificador.
Session_unregister('variable')	Abandona una variable sesión.

Vamos a crear un ejemplo de un contador de visitas en una página, si es la primera vez que entramos lo inicializa a 1, en caso contrario, lo incrementa en 1.

```
<? session_start();
if (!isset($_SESSION["cuenta_paginas"])) {
```

```

    $_SESSION["cuenta_paginas"] = 1;
}else{
    $_SESSION["cuenta_paginas"]++;
}
?>
<html>
<head>
<title>Contar páginas vistas por un usuario en toda su sesión</title>
</head>
<body>
<?
echo "Desde que entraste has visto " . $_SESSION["cuenta_paginas"] . " páginas";
?>
<br>
<br>
<a href="otracuenta.php">Ver otra página</a>
</body>
</html>

```

Para ver si una variable de sesión ha sido creada podemos utilizar la función *isset()* pasándole como parámetro la variable que deseamos saber si ha sido creada. Si no fue creada anteriormente, simplemente la creamos (inicializando su valor a 1). Si fue creada, pues que tenemos que incrementar en 1. Eso se consigue con este código:

```

if (!isset($_SESSION["cuenta_paginas"])) {
    $_SESSION["cuenta_paginas"] = 1;
} else {
    $_SESSION["cuenta_paginas"]++;
}

```

Ahora lo que se supone que tenemos son otras páginas en el mismo sitio que también tienen que incrementar en 1 el contador de páginas vistas por ese usuario en esa sesión.

En cualquier momento, si deseamos mostrar el número de páginas vistas en un momento dado, tendríamos que hacer algo como esto:

```

<?
echo "Has visto " . $_SESSION["cuenta_paginas"] . " páginas, contando actualizaciones
de páginas";
?>

```

ACTIVIDADES 13.46



- Partimos de una página donde se nos pide un usuario y una contraseña. Enviando los datos por POST y comprobamos que el usuario y contraseña sean "usuario" "usuario". Una vez hemos autenticado al usuario en el sistema, se inicia la sesión y, en ese momento, nos aparecerá un enlace a otra página web en la que deberemos comprobar que el usuario ha sido autenticado en el sistema, mostrando esta página una sección exclusiva para los usuarios registrados. Si el usuario ha accedido a esta página sin registrarse, deberá mostrar un enlace a la página principal para que se identifique correctamente.

13.30 PROBLEMAS DE AUTENTICACIÓN DE USUARIOS

Cuando estamos elaborando páginas web se nos puede presentar el caso de que a ciertas páginas o secciones de la web no deban acceder usuarios no autorizados. Vamos a resolver este problema utilizando variables de sesión, tendremos una página en nuestra web a la que no será posible acceder a menos que nos hayamos identificado y la variable de sesión haya sido a su vez inicializada.

Para nuestro ejemplo vamos a utilizar 2 páginas PHP: La primera de ellas pide un usuario y una contraseña y comprueba que son correctas, en caso afirmativo nos proporciona un enlace para ir a una parte privada, solo accesible a usuarios registrados. Si nos equivocamos de usuario y contraseña nos dará un enlace para volver a introducir la información.

```
<?
/* login.php */
session_start();
if (isset($_POST['enviar']))
{
/* hemos pulsado enviar en nuestra pagina */
$usuario=$_POST['usuario'];
$contrasena=$_POST['contrasena'];
if (($usuario=='alumno') and ($contrasena=='alumno'))
{
echo "usuario correcto";
echo "<br>";
$_SESSION['usuario']=$usuario; /* la variable de sesion es inicializada */
echo "<a href='privada.php'>Pulsa aqui para ir a la zona privada</a>";
}
else
{
echo "usuario incorrecto";
echo "<br>";
echo "<a href=''>Pulsa aqui para volver a intentarlo</a>";
}
}
else
{
/* no hemos pulsado todavia enviar en nuestra pagina */
?>
<form name="login" method="POST" action="">
usuario
<input type="text" name="usuario">
<br>
contrase&ntilde;a
<input type="password" name="contrasena">
<br><br>
<input type="submit" name="enviar" value="enviar">
```

```
</form>
<?
}
?>
```

usuario

contraseña

Figura 13.14. Autenticación

usuario incorrecto
[Pulsa aqui para volver a intentarlo](#)

Figura 13.15. Autenticación

usuario correcto
[Pulsa aqui para ir a la zona privada](#)

Figura 13.16. Autenticación

La segunda página comprueba si hemos accedido de forma legal, en dicho caso muestra bienvenido a mi página. En caso contrario nos invita a que nos identifiquemos como es debido.

```
<?
/* privada.php */
session_start();
if ((isset($_SESSION['usuario'])))
{
/* estamos de forma legal en la pagina podremos acceder a los servicios de la pagina*/
echo "bienvenido a la pagina";
}
else
{
/* si estamos aquí es porque la variable de sesion no tiene valor */
echo "no deberias estar aqui";
echo "<br>";
echo "<a href='login.php'>Accede como es debido</a>";
}
?>
```

bienvenido a la pagina

Figura 13.17. Autenticación

no deberias estar aqui
[Accede como es debido](#)

Figura 13.18. Autenticación

ACTIVIDADES 13.47



- Partiendo del caso práctico anterior, en la segunda página (una vez identificados), añadir un enlace que nos cierre la sesión y nos proporcione un enlace para la página inicial (aquella en la que teníamos que poner el nombre de usuario y contraseña).

ACTIVIDADES 13.48

- » Hacer una página que se comporte de manera distinta si el usuario está autenticado en el sistema o no (ha iniciado la sesión). Si el usuario no está identificado mostrará un enlace a otra página para proceder a la autenticación, mediante un usuario y una contraseña, una vez autenticado en el sistema deberá volver a la página inicial. Si el usuario esta autenticado mostrará unos enlaces de descarga.

13.31 MECANISMOS PARA MANTENER EL ESTADO ENTRE CONEXIONES

En PHP las conexiones que se mantienen pueden tener tres estados, *Normal* (0), *Aborted* (1) y *Timeout* (2).

En un *script* normal, el estado es **NORMAL**, cuando el cliente desconecta, el estado pasa a ser **ABORTED** y si el límite impuesto por PHP-imposed ha transcurrido, (*set_time_limit()*, el tiempo por defecto es 30 segundos) el estado es **TIMEOUT**.

Una función muy útil para estos casos, es *connection_status()* que devuelve el estado de la conexión.



RESUMEN DEL CAPÍTULO

Estamos ya casi terminando nuestro camino, estos dos últimos capítulos están dedicados a uno de los lenguajes de programación web orientada a servidor más utilizados, extendidos, vanagloriados y admirados de la actualidad, el lenguaje PHP.

¿Qué decir de PHP? Como lenguaje de servidor su potencia no tiene límite, nos permite hacer casi cualquier cosa y crear aplicaciones robustas y seguras con una curva de aprendizaje relativamente corta. De hecho, actualmente, no solo se limita a aplicaciones de servidor, sino que puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Quizás una de sus máximas cualidades es lo que vamos a estudiar en el capítulo siguiente: la interconexión con los sistemas gestores de bases de datos.

Con respecto a los aspectos negativos, ha habido ataques principalmente porque no es un lenguaje orientado a objetos (aunque los soporte) y al ser un lenguaje orientado al servidor los desarrollos hechos en PHP resultan en algunas ocasiones un poco lentos.



EJERCICIOS PROPUESTOS

- 1. ¿Qué diferencias principales encuentra entre los lenguajes orientados al cliente como JavaScript y los lenguajes orientados al servidor como PHP?
- 2. ¿Qué ventajas cree que tiene que en los lenguajes orientados al servidor la mayor parte de la carga del trabajo la realice el propio servidor?
- 3. PHP por sí solo no es capaz de recoger información proporcionada directamente por el usuario, ¿sobre qué lenguaje se apoya para poder tener interactividad con el usuario?
- 4. ¿Qué elementos principales del lenguaje sobre el que se apoya PHP para la interacción con el usuario utilizamos para enviar información por parte del usuario al servidor?
- 5. Algunos navegadores por defecto bloquean el código JavaScript, ¿por qué no ocurre lo mismo con el código PHP?



TEST DE CONOCIMIENTOS

- 1 PHP fue creado en:
 - a) 1982.
 - b) 1990.
 - c) 1992.
 - d) 1994.
 - e) 2000.
 - f) 2011.
- 2 Para mostrar texto por pantalla en PHP podemos utilizar:
 - a) Write.
 - b) Echo.
 - c) Muestra.
 - d) Todas son ciertas.
 - e) Todas son falsas.
- 3 Para concatenar cadenas de texto debo utilizar:
 - a) +
 - b) *
 - c) .
 - d) La función *concat*.
 - e) Todas son ciertas.
 - f) Todas son falsas.
- 4 Para finalizar una sesión iniciada y destruir todas las variables de sesión que existan en el sistema debo utilizar:
 - a) *Poweroff*.
 - b) *system_delete*.
 - c) *session_destroyer*.
 - d) *session_start*.
 - e) Todas son ciertas.
 - f) Todas son falsas.
- 5 El método de envío de información en el que todos los datos de la petición van a ser visibles desde la barra de direcciones del navegador, es el método:
 - a) POST.
 - b) REQUEST.
 - c) GET.
 - d) *session_start*.
 - e) Todas son ciertas.
 - f) Todas son falsas.

14

Acceso a bases de datos desde lenguajes de *script* de servidor

OBJETIVOS DEL CAPÍTULO

- ✓ Conectar a Sistemas Gestores de Bases de Datos desde PHP.
- ✓ Seleccionar bases de datos para trabajar con ellas.
- ✓ Crear tablas dentro de las bases de datos.
- ✓ Insertar, eliminar y modificar información en las tablas de la base de datos.
- ✓ Consultar información almacenada en las tablas de la base de datos.

14.1 INTEGRACIÓN DE LOS LENGUAJES DE SCRIPT DE SERVIDOR CON LOS SISTEMAS GESTORES DE BASE DE DATOS

Como ya hemos visto, PHP es un lenguaje de programación precompilado y desplegado en nuestro navegador de Internet. Si bien es cierto que un lenguaje basado en *scripts* es mucho más sencillo de aprender y aplicar que uno precompilado, éste último es mucho más poderoso y ofrece muchas más posibilidades de desarrollo e integración que un lenguaje basado en *scripts*.

Ahora bien, lo que la integración implica la posibilidad de trabajar de la mano con otras herramientas de desarrollo, como la integración con las bases de datos. La mencionada integración ocurre de la siguiente forma:

- ✓ La aplicación se conecta con el motor de base de datos.
- ✓ La aplicación hace una consulta (u otro tipo de operación).
- ✓ La base de datos proporciona la información solicitada.
- ✓ La aplicación “recoge” la información que el motor de base de datos proporciona.
- ✓ La aplicación maneja la información presentada.
- ✓ La aplicación presenta un resultado en nuestra pantalla.

Muy bien pero, ¿qué es un motor de base de datos? Un motor de base de datos no es más que una aplicación que maneja bases de datos (es decir, sirve de interfaz entre la aplicación y la base de datos como tal) a través de un lenguaje de consultas para comunicarse con los archivos que guardan los datos

14.2 CONEXIÓN A BASES DE DATOS MYSQL CON PHP

El sistema de acceso y manipulación de bases de datos desde PHP es similar al de otros lenguajes de *script*: establece la conexión con la base de datos, ejecuta las sentencias de consulta o modificación y, finalmente, cierra la conexión.

PHP soporta compatibilidad con accesos a múltiples sistemas de bases de datos. Sin embargo, el modo de programación sobre cada tipo de base de datos, no presenta la misma nomenclatura. PHP utiliza funciones de nombre genérico, pero precedidas normalmente por el nombre del sistema de base de datos, de modo que las funciones que ofrece el lenguaje para el acceso a cada tipo de base de datos son diferentes.

Para establecer la conexión con una base de datos MySQL desde PHP, se utiliza la función:

```
mysql_connect("NombreHost", "Usuario", "Contraseña")
```

Cuyos parámetros son opcionales, e indican el nombre del servidor (o IP del mismo), '*NombreHost*', donde se encuentra la base de datos, el '*Usuario*' de acceso a la base de datos y la '*Contraseña*' de acceso. En caso de éxito la función devuelve un identificador del enlace con el sistema de bases de datos.

Una vez establecida la conexión, hay que indicar al sistema, la base de datos con la que se desea trabajar, usando la función:

```
mysql_select_db("NombreBaseDatos")
```

Y, finalmente, para cerrar la conexión utilizamos:

```
mysql_close($conexion)
```

Donde debemos pasarle como parámetro el enlace con la conexión inicialmente establecida. Ejemplo:

```
<?php
# Conectamos con la base de datos MySQL
$conexion = mysql_connect("NombreHost", "Usuario", "Contraseña") or die("No ha podido
realizarse la conexión");
#Seleccionamos la base de datos a utilizar
mysql_select_db("NombreBaseDatos") or die("No ha podido seleccionarse la base de datos");
#Cerramos la conexión con la base de datos
mysql_close($conexion);
?>
```

PRÁCTICA 14.1



➤ Vamos a conectar con una base de datos local.

```
$MySQL_host = "localhost";
$MySQL_database = "prueba";
$MySQL_user = "juan";
$MySQL_password = "xxxxxxx";
$conexion = mysql_connect($MySQL_host,$MySQL_user,$MySQL_password) or die("No ha
podido realizarse la conexión");
mysql_select_db($MySQL_database) or die("No ha podido seleccionarse la base de
datos");
```

PRÁCTICA 14.2



➤ Vamos a conectar con una base de datos creada en un servidor gratuito en Internet.

```
$MySQL_host = "MySQL2.000webhost.com";
$MySQL_database = "prueba";
$MySQL_user = "a5953119";
$MySQL_password = "xxxxxxx";
mysql_connect($MySQL_host,$MySQL_user,$MySQL_password) or die("No ha podido
realizarse la conexión");
mysql_select_db($MySQL_database) or die("No ha podido seleccionarse la base de
datos");
```



Podemos guardar los datos anteriores en un fichero *conexión.inc.php* que realice la conexión a la base de datos y la selección de la base de datos.

ACTIVIDADES 14.1



➤ Crear un fichero *conexión.inc.php* para la conexión con su servidor y su base de datos.

14.3 CREACIÓN DE BASES DE DATOS EN MYSQL CON PHP

Para crear una base de datos nueva, en primer lugar debemos conectar con el servidor de bases de datos. Para crear una base de datos en MySQL desde PHP debemos utilizar la función *mysql_create_db*("nombre de la base de datos a crear"). A continuación podemos ver un ejemplo de cómo funciona esta función:

```
<?
mysql_connect("maquina","usuario","password") or die("No se ha podido conectar con la
base de datos");
mysql_create_db("basedatos") or die("No se ha podido crear la base de datos");
?>
```

ACTIVIDADES 14.2



➤ Crear una base de datos en nuestro servidor que hemos configurado en Práctica 14.1.

Una vez creada la base de datos, la seleccionamos utilizando *mysql_select_db*("nombre de nuestra base de datos"), este paso es obligatorio para poder utilizar una base de datos en nuestro sistema, por tanto, para poder utilizar las tablas de una base de datos, previamente debemos seleccionarla.

ACTIVIDADES 14.3



➤ Seleccionar la base de datos creada en nuestro servidor en la Práctica 14.2.

PRÁCTICA 14.3



- Vamos a crear un fichero con una función para conectar con el gestor de bases de datos, otra función para crear bases de datos, y otra para seleccionar una base de datos ya existente. A este fichero le vamos a llamar *funcionesbd.inc.php*.

```
<?
function conecta()
{
    $MySQL_host = "localhost";
    $MySQL_user = "juan";
    $MySQL_password = "xxxxxxx";
    mysql_connect($MySQL_host,$MySQL_user,$MySQL_password) or die("No ha podido
    realizarse la conexión con la base de datos");
}
function crea($basedatos)
{
    mysql_create_db("$basedatos") or die("No se ha podido crear la base de datos");
}
function selecciona($basedatos)
{
    mysql_select_db("$basedatos") or die("Error en la selección de la base de datos");
}
?>
```

14.4 CREACIÓN DE TABLAS EN MYSQL CON PHP

Para la creación de tablas en MySQL, podemos utilizar estos dos comandos:

```
CREATE TABLE tabla (campo1, campo2, ..);
CREATE TABLE IF NOT EXISTS tabla (campo1, campo2, ..);
```

Ejemplo de CREATE TABLE en MySQL: Vamos a crear una tabla llamada *prueba* dentro de la base de datos anterior que tiene un campo llamado número de tipo entero, un campo nombre que es de tipo texto, un campo *dirección*, que es de tipo texto, y un campo *fecha* que es de tipo fecha. La clave primaria de la tabla será número:

```
<?
include "funcionesbd.inc.php";
conecta();
crea("bdprueba");
selecciona("bdprueba");
$tabla="CREATE TABLE prueba (numero INT PRIMARY KEY, nombre VARCHAR(200), direccion
VARCHAR(200), fecha DATE);";
MySQL_query($tabla) or die("No se ha podido crear la tabla");
?>
```

PRÁCTICA 14.4



➤ Vamos a crear una tabla dentro de la base de datos *prueba*. La tabla tiene las siguientes características:

	producto
🔑	codproducto : varchar(20)
📄	descripcion : varchar(200)
#	precio : double

Figura 14.1

```
<?
include "funcionesbd.inc.php";
conecta();
selecciona("bdprueba");
$tabla="CREATE TABLE producto (codproducto VARCHAR(20) PRIMARY KEY, descripción
VARCHAR(200), precio DOUBLE);";
mysql_query($tabla) or die("No se ha podido crear la tabla");
?>
```

ACTIVIDADES 14.4



➤ Crear dentro de la base de datos seleccionada en la Actividad 14.3 las siguientes tablas:

<table> <tr> <th></th><th>cliente</th></tr> <tr> <td>🔑</td><td>codcliente : varchar(20)</td></tr> <tr> <td>📄</td><td>nombre : varchar(100)</td></tr> <tr> <td>📄</td><td>apellidos : varchar(200)</td></tr> <tr> <td>📄</td><td>direccion : varchar(200)</td></tr> <tr> <td>📄</td><td>telefono : varchar(20)</td></tr> <tr> <td>📄</td><td>email : varchar(100)</td></tr> </table>		cliente	🔑	codcliente : varchar(20)	📄	nombre : varchar(100)	📄	apellidos : varchar(200)	📄	direccion : varchar(200)	📄	telefono : varchar(20)	📄	email : varchar(100)	<table> <tr> <th></th><th>producto</th></tr> <tr> <td>🔑</td><td>codproducto : varchar(20)</td></tr> <tr> <td>📄</td><td>descripcion : varchar(200)</td></tr> <tr> <td>#</td><td>precio : double</td></tr> </table>		producto	🔑	codproducto : varchar(20)	📄	descripcion : varchar(200)	#	precio : double
	cliente																						
🔑	codcliente : varchar(20)																						
📄	nombre : varchar(100)																						
📄	apellidos : varchar(200)																						
📄	direccion : varchar(200)																						
📄	telefono : varchar(20)																						
📄	email : varchar(100)																						
	producto																						
🔑	codproducto : varchar(20)																						
📄	descripcion : varchar(200)																						
#	precio : double																						
<table> <tr> <th></th><th>albaran</th></tr> <tr> <td>🔑</td><td>numalbaran : int(11)</td></tr> <tr> <td>📄</td><td>codcliente : varchar(20)</td></tr> <tr> <td>📅</td><td>fecha : date</td></tr> </table>		albaran	🔑	numalbaran : int(11)	📄	codcliente : varchar(20)	📅	fecha : date	<table> <tr> <th></th><th>albaranlinea</th></tr> <tr> <td>🔑</td><td>numalbaran : int(11)</td></tr> <tr> <td>🔑</td><td>numlinea : int(11)</td></tr> <tr> <td>📄</td><td>codproducto : varchar(20)</td></tr> <tr> <td>#</td><td>cantidad : int(11)</td></tr> </table>		albaranlinea	🔑	numalbaran : int(11)	🔑	numlinea : int(11)	📄	codproducto : varchar(20)	#	cantidad : int(11)				
	albaran																						
🔑	numalbaran : int(11)																						
📄	codcliente : varchar(20)																						
📅	fecha : date																						
	albaranlinea																						
🔑	numalbaran : int(11)																						
🔑	numlinea : int(11)																						
📄	codproducto : varchar(20)																						
#	cantidad : int(11)																						

Figura 14.2

14.5 CREACIÓN DE VISTAS EN MYSQL CON PHP. PROCEDIMIENTOS ALMACENADOS

Una vista puede considerarse como una tabla virtual. Podemos filtrar los datos que nos interese que sean mostrados por la vista y ocultar otra serie de datos. Si actualizamos la vista, la tabla original se actualiza. Si actualizamos la tabla la vista también queda actualizada. En MySQL las vistas se crean con la sentencia CREATE VIEW, acompañada de la sentencia SELECT que proporciona la definición de la vista. Puede estar dirigida a tablas de la base o a otras vistas.

```
$vista="CREATE VIEW test SELECT * FROM producto;";  
mysql_query($vista);
```

Si queremos modificar una vista existente debemos utilizar REPLACE VIEW.

```
$vista2="REPLACE VIEW test SELECT * FROM clientes;";  
mysql_query($vista2);
```

ACTIVIDADES 14.5



- Crear una vista de la tabla cliente que solo tenga los campos *código*, *nombre*, *apellidos* y *email*.

Los procedimientos almacenados y funciones son nuevas funcionalidades de la versión de MySQL 5.0. Un procedimiento almacenado es un conjunto de comandos SQL que pueden almacenarse en el servidor. Una vez que se hace, los clientes no necesitan relanzar los comandos individuales pero pueden en su lugar referirse al procedimiento almacenado.

Algunas situaciones en que los procedimientos almacenados pueden ser particularmente útiles:

- Cuando múltiples aplicaciones cliente se escriben en distintos lenguajes o funcionan en distintas plataformas, pero necesitan realizar la misma operación en la base de datos.
- Cuando la seguridad es muy importante. Los bancos, por ejemplo, usan procedimientos almacenados para todas las operaciones comunes. Esto proporciona un entorno seguro y consistente y los procedimientos pueden asegurar que cada operación se *loguea* apropiadamente. En tal entorno, las aplicaciones y los usuarios no obtendrían ningún acceso directo a las tablas de la base de datos, solo pueden ejecutar algunos procedimientos almacenados.

Los procedimientos almacenados pueden mejorar el rendimiento ya que se necesita enviar menos información entre el servidor y el cliente. El intercambio que hay es que aumenta la carga del servidor de la base de datos ya que la mayoría del trabajo se realiza en la parte del servidor y no en el cliente. Considere esto si muchas máquinas cliente (como servidores web) se sirven a solo uno o pocos servidores de bases de datos.

Los procedimientos almacenados le permiten tener bibliotecas o funciones en el servidor de base de datos. Esta característica es compartida por los lenguajes de programación modernos que permiten este diseño interno, por ejemplo, usando clases. Usando estas características del lenguaje de programación cliente es beneficioso para el programador incluso fuera del entorno de la base de datos.

La implementación de MySQL de procedimientos almacenados está en progreso.

```
$procedimiento="CREATE PROCEDURE estudiantes_buscar(letra
CHAR(2)) SELECT * FROM estudiante WHERE nombre LIKE letra;";
mysql_query($procedimiento);
```

Para llamar al procedimiento debemos utilizar la función *call*:

```
echo "call estudiantes_buscar('p%');";
```

ACTIVIDADES 14.6



- Crear procedimiento almacenado llamado *productobuscar* que permita buscar los productos por descripción. Hacer también una llamada *call* a dicho procedimiento almacenado.

14.6 INSERCIÓN DE INFORMACIÓN EN TABLAS MYSQL CON PHP

La forma más directa de insertar una fila nueva en una tabla es mediante una sentencia INSERT. En la forma más simple de esta sentencia debemos indicar la tabla a la que queremos añadir filas y los valores de cada columna.

Las columnas de tipo cadena o fechas deben estar entre comillas sencillas o dobles, para las columnas numéricas esto no es imprescindible, aunque también pueden estar entrecomilladas.

PRÁCTICA 14.5



- Sobre la tabla creada en la anterior práctica, vamos a insertar los datos siguientes:

PRODUCTO		
codproducto	descripcion	precio
A0001	caja atx fuente 500 watt	25,30 €
A0002	placa base intel p5g41t-m lx	36,00 €
A0003	procesador intel celeron e3100	25,00 €

:

Figura 14.3

```
<?
include "funcionesbd.inc.php";
conecta();
selecciona("bdprueba");
$insertar="INSERT INTO PRODUCTO VALUES ('A0001','caja atx fuente 500 watt',25.30),
('A0002','placa base intel p5g41t-m lx',36.00),('A0003','procesador intel celeron
```



```
e3100',25.00);";
mysql_query($insertar) or die("No se ha podido crear la tabla");
?>
```

ACTIVIDADES 14.7



➤ Insertar en las tablas creadas, los siguientes datos.

CLIENTE					
codcliente	nombre	apellidos	direccion	telefono	email
a01	Agustín	Alonso Berzosa	c/ de la vida 2	601601601	pe@as.com
a02	Martha Yaneth	Bergaño Jaramillo	c/ de la vida 3	602602602	santiago@yahoo.es
b01	Carina Janneth	Collaguazo Lalangui	c/ de la vida 4	603603603	mio@tuyo.es
b02	Ramón	Díaz García	c/ de la vida 5	604604604	maria@yahoo.es
b03	María Francisca	Giménez Llagunes	c/ de la vida 6	605605605	chesca@gmail.com
b04	Isabel	Losilla Pérez	c/ de la vida 7	606606606	isable@hotmail.com
b05	El Bachir	Moujib Dadou	c/ de la vida 8	604604604	elbachir@gmail.com
b06	Sergio	Reyes Céspedes	c/ de la vida 9	606206205	sergio@yahoo.es
b07	José María	Soler González	c/ de la vida 10	606606606	josemaria@gmail.com

PRODUCTO		
codproducto	descripcion	precio
A0001	caja atx fuente 500 watt	25,30 €
A0002	placa base intel p5g41t-m lx	36,00 €
A0003	procesador intel celeron e3100	25,00 €
A0004	memoria 2gb pc-1333 ddr3	12,00 €
A0005	disco duro 500 gb sata3	60,00 €
A0006	lector dvd rw sata lg h22n	22,00 €

ALBARAN		
numalbaran	codcliente	fecha
1	b02	11/11/2011
2	b06	12/11/2011
3	a01	13/11/2011
4	b02	14/11/2011
5	a01	15/11/2011
6	b07	16/11/2011

ALBARANLINEA			
numalbaran	numlinea	codproducto	cantidad
1	1	A0001	3
1	2	A0002	2
2	3	A0004	3
2	4	A0002	12
3	5	A0004	1
4	6	A0002	1
4	7	A0004	2
4	8	A0004	5
5	9	A0002	6

Figura 14.4

14.7 RECUPERACIÓN DE LA INFORMACIÓN DE LA BASE DE DATOS MYSQL DESDE UNA PÁGINA PHP

Para la recuperación de los datos de una base de datos vamos a utilizar la instrucción SELECT, el resultado de la función SELECT se puede dividir en 2 apartados según el resultado devuelto:

14.7.1 CONSULTA CON VARIOS RESULTADOS

Solo con la ejecución de la consulta sobre la base de datos, no podemos presentar el resultado de la misma. Para poder mostrar información resultante de una consulta deberemos hacer uso de funciones complementarias. Una de las posibles es `mysql_fetch_array` que devuelve un *array* con el resultado de la consulta. El uso de `mysql_fetch_array` es el siguiente:

```
MySQL_query(consulta sql);
$array_resultado=MySQL_fetch_array($id_resultado);
```

Que devuelve un *array* con el resultado de la sentencia extraída identificada por el parámetro facilitado. Podemos recorrer el *array* resultado con *while* o con *for each*.

PRÁCTICA 14.6



- Recuperamos la información de la tabla producto de nuestra base de datos de ejemplo y separamos los datos de nuestra base de datos separados por “,”.

```
<?
include "funcionesbd.inc.php";
conecta();
selecciona("bdprueba");
$res="SELECT * FROM producto;";
$resultado=mysql_query($res);
while ($fila=mysql_fetch_array($resultado)) {
echo $fila['codigo'];
echo ", ";
echo $fila['descripcion'];
echo ", ";
echo $fila['pvp'];
echo ", ";
echo $fila['stock'];
echo "<br> ";
}
?>
```

14.7.2 CONSULTA CON UN SOLO RESULTADO

En este caso la consulta no devuelve un *array*, sino que obtenemos un solo resultado que puede ser almacenado en una variable sencilla (no un *array*). Para ello utilizamos la función *MySQL_result(MySQL_query,0)*.

```
$sql="SELECT descripcion FROM producto WHERE codigo='a0001'";  
$consulta=mysql_query($sql);  
$variable=mysql_result($consulta,0);
```

PRÁCTICA 14.7



➤ Hacemos una consulta que solo devuelve un resultado.

```
<?  
include "funcionesbd.inc.php";  
conecta();  
selecciona("bdprueba");  
$res2="SELECT descripción FROM producto WHERE codproducto='A0001'";  
$resultado2=mysql_query($res2);  
$descripcion=mysql_result($resultado2,0);  
echo $descripcion;  
?>
```

ACTIVIDADES 14.8



➤ Hacer las siguientes consultas:

- Mostrar todos los datos de los clientes que viven en la calle "de la vida".
- Mostrar todos los datos de los clientes cuyo código empiece por *a*.
- Mostrar todos los datos de los clientes cuyo *email* sea de Gmail.
- Mostrar código del cliente con nombre "Ramón" y apellidos "Díaz García"
- Mostrar el nombre y apellidos del cliente con código "b06".

14.8 MODIFICACIÓN DE LA INFORMACIÓN ALMACENADA. ACTUALIZACIONES Y BORRADOS

14.8.1 ACTUALIZAR CAMPOS DE LA TABLA

Podemos modificar valores de las filas de una tabla usando la sentencia *UPDATE*. En su forma más simple, los cambios se aplican a todas las filas, y a las columnas que especifiquemos.

Por ejemplo, podemos aumentar en un 10% la población de todas las ciudades de la tabla *ciudad3* usando esta sentencia:

```
$res2="UPDATE ciudad3 SET poblacion=poblacion*1.10;";
mysql_query($res2) or die("No se ha podido actualizar la tabla");
```

Si lo que queremos es actualizar un campo determinado y no todos deberemos utilizar la cláusula WHERE. Usando esta cláusula podemos establecer una condición. Solo las filas que cumplan esa condición serán actualizadas:

```
$res2="UPDATE ciudad5 SET poblacion=poblacion*1.03 WHERE nombre='Roma'";
mysql_query($res2) or die("No se ha podido actualizar la tabla");
```

En este caso solo hemos aumentado la población de las ciudades cuyo nombre sea *Roma*. Las condiciones pueden ser más complejas. Existen muchas funciones y operadores que se pueden aplicar sobre cualquier tipo de columna y también podemos usar operadores booleanos como AND u OR.

PRÁCTICA 14.8



➤ Sobre la tabla *productos* de la Base de Datos *prueba* modificamos la descripción del campo con código A0002 por "Placa Base Asus P8H61-M LX" y su precio por 48 €.

<?

```
include "funcionesbd.inc.php";
conecta();
selecciona("bdprueba");

$mod1="UPDATE producto SET descripcion='Placa Base Asus P8H61-M LX' WHERE
codproducto='A0002'";

mysql_query($mod1) or die("No se ha podido actualizar la tabla");

$mod2="UPDATE producto SET precio=48.00 WHERE codigo='A0002'";

mysql_query($mod2) or die("No se ha podido actualizar la tabla");

?>
```

14.8.2 ELIMINAR CAMPOS DE UNA TABLA

Para eliminar filas se usa la sentencia DELETE. La sintaxis es muy parecida a la de UPDATE.

```
$res2="DELETE FROM ciudad3;";
MySQL_query($res2) or die("No se ha podido actualizar la tabla");
```

De este modo se eliminan todas las filas de la tabla. Pero es más frecuente que solo queramos eliminar ciertas filas que cumplan determinadas condiciones. La forma más normal de hacer esto es usar la cláusula WHERE:

```
$res2="DELETE FROM ciudad5 WHERE clave=2;";
MySQL_query($res2) or die("No se ha podido actualizar la tabla");
```

PRÁCTICA 14.9

- Sobre la tabla *productos* de la base de datos *prueba* vamos a eliminar el producto con código A0003.

```
<?
include "funcionesbd.inc.php";
conecta();
selecciona("bdprueba");
$res2="DELETE FROM producto WHERE codproducto='A0003'";
mysql_query($res2) or die("No se ha podido actualizar la tabla");
?>
```

ACTIVIDADES 14.9

- Hacer las modificaciones sobre los campos de la tabla *CLIENTE*:
- El cliente con código b01 hay que modificar el nombre de "Carina Janneth" por "Karina Yaneth".
 - El cliente con código b03 ha cambiado su teléfono móvil por 666666666.
 - El cliente b05 se ha dado de baja y por tanto debe ser eliminado de la tabla de clientes.
 - El cliente b04 ha cambiado su *email* por *isabel@hotmail.com*.

ACTIVIDADES 14.10

- Hacer las modificaciones sobre los campos de la tabla *PRODUCTO*:
- El producto A0003 ha variado su precio, pasando a ser 20,00 €.
 - El producto A0005 ha variado su precio, pasando a ser de 150,00 €.
 - El producto A0006 ha sido dado de baja, por tanto debe eliminarse de la tabla.

14.9 FUNCIONES ODBC

ODBC que es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué sistema gestor de bases de datos almacene dichos datos.

Las siglas ODBC significan en inglés *Open DataBase Connectivity* en español se denomina como Conectividad Abierta de Bases de Datos, la cual se logra al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS. El propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda.

Para que esto funcione tanto la aplicación como el DBMS deben ser compatibles con ODBC, esto significa que la aplicación debe ser capaz de producir comandos ODBC y el DBMS debe ser capaz de responder a ellos. Desde la versión 2.0 el estándar soporta SAG y SQL.

En algunas ocasiones se nos presentan diferentes problemas a solucionar, como el de desarrollar una aplicación con la cual podamos conectarnos a diferentes bases de datos como: Access, dBase, entre otras. Una forma de resolver este problema es usando ODBC. En el siguiente ejemplo agregaremos un DSN dentro del ODBC para realizar la conexión a una base de datos en Access:

Iniciamos abriendo nuestro ODBC Data Source Administrator, luego nos situamos en la pestaña System DSN. (Figura 14.5).

Posicionados en la pestaña **System DSN** agregaremos uno nuevo, seleccionando el botón **Add/Agregar** (Figura 14.6).

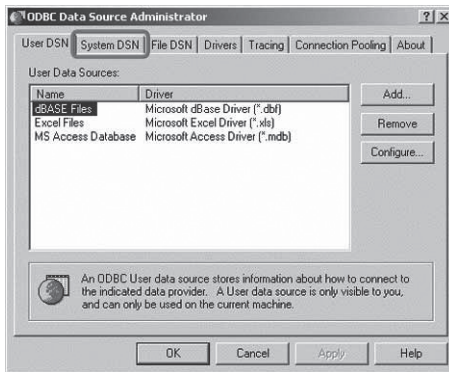


Figura 14.5. System DNS

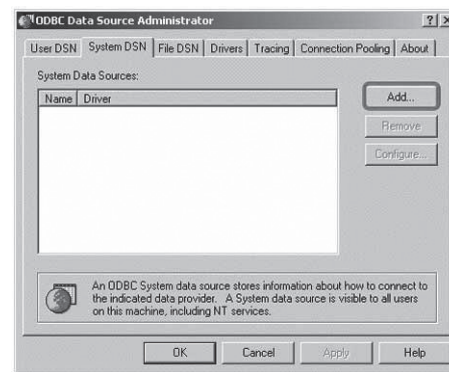


Figura 14.6. Add...

En este caso seleccionamos el Driver de Microsoft Access (*.mdb) y damos clic en el botón **Finish/Finalizar** (Figura 14.7).

Introducimos el *Data Source Name* tomando en cuenta que este nombre llamará a nuestro DSN creado en ODBC. En este ejemplo se llamará: prueba. La descripción (opcional). Luego hacemos clic en el botón **Select/Seleccionar**, donde buscaremos nuestra base de datos, en el ejemplo *db.mbd* y seleccionemos el botón **OK** (Figura 14.8).

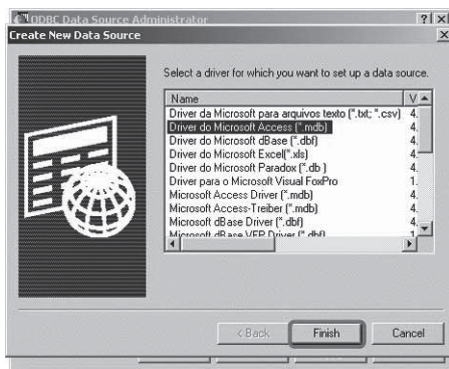


Figura 14.7. Driver de Access

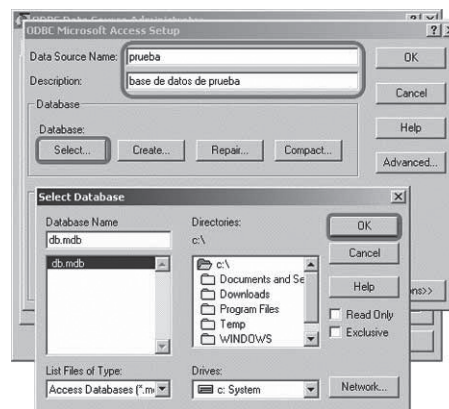


Figura 14.8. DNS

Después seleccionamos nuestra base de datos, con todos los datos correctos, seleccionamos el botón **OK** (Figura 14.9).

A continuación vemos que se ha agregado nuestro DSN, con nombre *prueba* y el Driver “Driver do Microsoft Access” (*.mdb). Ahora seleccionemos el botón **OK** (Figura 14.10).

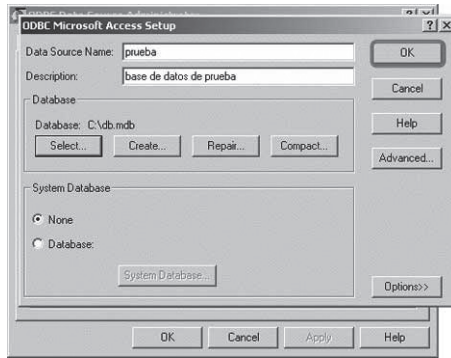


Figura 14.9. Seleccionamos nuestra BD

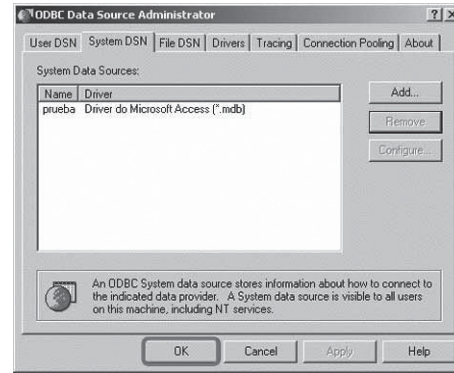


Figura 14.10. Ok

Una vez creado nuestro DSN en el ODBC, veremos el siguiente código PHP, el cual nos permitirá conectarnos a nuestro ODBC y realizar consultas a la base de datos:

```
<?
$dsn = "prueba";
//debe ser de sistema no de usuario
$usuario = "";
$clave="";
//realizamos la conexion mediante odbc
$cid=odbc_connect($dsn, $usuario, $clave);
if (!$cid){
    exit("<strong>Ya ocurrido un error tratando de conectarse con el origen de datos.</strong>");
}
// consulta SQL a nuestra tabla "usuarios" que se encuentra en la base de datos "db.mdb"
$sql="Select * from usuarios";
// Generamos la tabla mediante odbc_result_all(); utilizando borde 1
$result=odbc_exec($cid,$sql)or die(exit("Error en odbc_exec"));
print odbc_result_all($result,"border=1");
?>
```

Esto también nos ayudará a la hora de usar otros tipos de bases de datos, solo necesitaremos modificar el *driver* que utilizaremos en nuestro ODBC, sin necesidad de cambiar nuestro código en PHP.

14.10 ADODB

PHP está especialmente diseñado para la creación de sitios webs dinámicos. Para crear estos sitios normalmente se utiliza algún tipo de base de datos desde donde obtenemos la información que queremos mostrar, ya sean noticias, preguntas y respuestas de un foro u otro tipo de información dinámica. Si usamos php normalmente usaremos MySQL como base de datos para iniciar nuestro sitio, el problema aparece cuando nuestro proyecto crece tanto que necesita hacer uso de otro tipo de base de datos más robusta.

Desafortunadamente el acceso en PHP cada base de datos es muy diferente. Para conectarnos a MySQL, debemos usar `mysql_connect()`; cuando decida cambiar a Oracle o Microsoft SQL Server, debe usar `oci_logon()` o `mssql_connect()`, respectivamente. Lo peor es que también los parámetros de cada función son diferentes.

Una librería de abstracción de los datos como ADODB es lo que Ud. necesita si desea asegurar en gran medida la portabilidad de su aplicación. Provee una serie de funciones comunes para comunicarse con las distintas bases de datos.

ADODB viene de *Active Data Objects DataBase* (no muy original pero efectivo). ADODB actualmente soporta MySQL, PostgreSQL, Oracle, Interbase, Microsoft SQL Server, Access, FoxPro, Sybase, ODBC y ADO. Puede obtener ADODB en <http://adodb.sourceforge.net/#download>.

14.10.1 EJEMPLO MYSQL

```
$db = mysql_connect("localhost", "root", "password");
mysql_select_db("mydb");
$result = mysql_query("SELECT * FROM employees");
while ($fields = mysql_fetch_array($result)) {
    echo $fields['campo1'];
    echo $fields['campo2'];
}
print "<br>";
}
```

14.10.2 AQUÍ SU EQUIVALENTE EN ADODB

```
include("adodb.inc.php");
$db = NewADOConnection('mysql');
$db->Connect("localhost", "root", "password", "mydb");
$result = $db->Execute("SELECT * FROM employees");
while (!$result->EOF) {
    print $result->fields['campo1'];
    print "<br>";
    print $result->fields['campo2'];
    print "<br>";
    $result->MoveNext();
}
```

Ahora portarlo a Oracle es tan simple como cambiar la segunda línea a `NewADOConnection('oracle')`. Vamos a mirar cada parte del código.

14.10.3 CONECTANDO A LA BASE DE DATOS

```
include("adodb.inc.php");
$db = NewADOConnection('mysql');
$db->Connect("localhost", "root", "password", "mydb");
```

La conexión puede parecer algo más complicada que en MySQL pero se explica en que ADODB es totalmente orientado a objetos. Si no está familiarizado con la programación orientada a objetos no se preocupe porque todo queda oculto en la función *NewADOConnection()*.

Para ahorrar memoria solo se carga las funciones específicas de la base de datos que vamos a utilizar. Cada *driver* tiene un archivo distinto que se carga mediante un *include* automáticamente.

Una vez cargado el driver que vamos a utilizar mediante la función *NewADOConnection()* nos conectamos a la base de datos usando *\$db->Connect()*.

14.10.4 EJECUTANDO SQL

```
$result = $db->Execute("SELECT * FROM employees");
```

Para enviar una sentencia sql al motor de base de datos se utiliza la función *Execute()*. Esta función devuelve un objeto *recordset* si la ejecución fue correcta o un *false* si hubo algún error.

Existen dos objetos principales en ADODB. El objeto *connection* y el objeto *recordset*. ¿Cuándo usar cada uno?

El objeto *connection* (*\$db*) es responsable de la conexión a la base de datos, formateo de las SQL y las consultas al servidor. El objeto *recordset* (*\$result*) es responsable de la obtención de los resultados y el formateo de los resultados.

14.10.5 OBTENIENDO DATOS

```
while (!$result->EOF) {
    print $result->fields['campo1'];
    print "<br>";
    print $result->fields['campo2'];
    print "<br>";
    $result->MoveNext();
}
```

El proceso para obtener datos es como el proceso para leer desde un fichero. Para cada línea observamos si hemos llegado al final del fichero (EOF). Mientras no lleguemos al final del fichero seguimos leyendo y moviéndonos a la siguiente línea (*movenext*).

El array *\$result->fields[]* es generado por ADODB ante cada petición de *movenext()*. Podemos obtener array asociativos o indexados por el número del campo.

14.10.6 ADOCONNECTION

El objeto que realiza la conexión a la base de datos, ejecuta las sentencias SQL y tiene otro *set* de funciones para estandarizar el formato de sentencias SQL como la concatenación de cadenas o formatos de fechas.

14.10.7 OTRAS FUNCIONES ESPECIALES

`$recordset->Move($pos)` mueve el *recordset* a una fila en particular. ADODB soporta avance de filas en todas las bases de datos. Algunas bases de datos no soportan retroceso en el *recordset*. Puede de todas maneras simular este comportamiento usando una caché.

- **`$recordset->RecordCount()`** retorna el total de filas obtenidas en el *recordset*. Algunas bases de datos no soportan esto y devuelven un -1.
- **`$recordset->GetArray()`** coloca el resultado del *recordset* en un *array*.
- **`rs2html($recordset)`** esta función genera una tabla HTML con el resultado del *recordset*.

```
include('adodb.inc.php');
include('tohtml.inc.php'); /* includes the rs2html function */
$conn = &ADONewConnection('mysql');
$conn->PConnect('localhost','userid','password','database');
$rs = $conn->Execute('select * from table');
rs2html($rs); /* recordset to html table */
```

Más funciones útiles pueden encontrarse en la página web <http://adodb.sourceforge.net/#docs>.



RESUMEN DEL CAPÍTULO

Concluimos PHP este libro. Como se comentó en el capítulo anterior, una de las mayores y mejores características de PHP es el acceso a bases de datos de forma sencilla y rápida.

En este capítulo hemos visto cómo conectar con MySQL, cómo seleccionar bases de datos dentro de él, cómo crear tablas, insertar datos, actualizar dichos datos y cómo eliminarlos de la tabla.

Por si acaso nuestro gestor de bases de datos favorito no está soportado nativamente por PHP, también hemos visto cómo conectar con ellos mediante ODBC.



EJERCICIOS PROPUESTOS

- 1. Aparte de MySQL, PHP soporta otros gestores de bases de datos. Investigue por la red cuáles son.
- 2. ¿Qué ventajas encuentra en PHP con respecto al acceso a datos que no tienen otros lenguajes como Java o C++?
- 3. Busque información en Internet sobre un *framework* basado en PHP y MySQL llamado Symfony.
- 4. Busque información en Internet sobre un *framework* basado en PHP y MySQL llamado Zend Framework.
- 5. Obtenga más información sobre ADOBD de su página principal o de otras.



TEST DE CONOCIMIENTOS

- 1 Para conectar con un servidor de bases de datos MySQL debemos utilizar la función:
 - a) *mysql_query*.
 - b) *mysql_select_db*.
 - c) *mysql_close*.
 - d) *mysql_connect*.
 - e) *mysql_fetch_array*.
 - f) Todas son falsas.
- 2 Una vez hemos conectado, para seleccionar una base de datos MySQL en particular debemos analizar la función:
 - a) *mysql_query*.
 - b) *mysql_select_db*.
 - c) *mysql_close*.
 - d) *mysql_connect*.
 - e) *mysql_fetch_array*.
 - f) Todas son falsas.
- 3 Para dar nuestra conexión con el sistema gestor de bases de datos por concluida debemos alizar la función:
 - a) *mysql_query*.
 - b) *mysql_select_db*.
 - c) *mysql_close*.
 - d) *mysql_connect*.
 - e) *mysql_fetch_array*.
 - f) Todas son falsas.
- 4 Para ejecutar una sentencia sobre la base de datos MySQL, debemos utilizar la función:
 - a) *mysql_query*.
 - b) *mysql_select_db*.
 - c) *mysql_close*.
 - d) *mysql_connect*.
 - e) *mysql_fetch_array*.
 - f) Todas son falsas.
- 5 Cuando el resultado de una consulta devuelve muchos valores de forma organizada, para poder utilizarlos debemos utilizar la función:
 - a) *mysql_query*.
 - b) *mysql_select_db*.
 - c) *mysql_close*.
 - d) *mysql_connect*.
 - e) *mysql_fetch_array*.
 - f) Todas son falsas.

Índice Alfabético

A

ADODB, 372
Apache, 44, 56
Autenticación, 353

B

Back-end, 73
Blog, 82
Blogger, 85

C

Cliente servidor, 21
CMS, 72

D

DOM, 300
Dreamweaver, 315

E

Eclipse, 315
Else, 328
Elseif, 329
Eventos, 308
Excepciones, 344

F

For, 332
Formularios, 278, 345
Front-end, 73
FTP, 25

G

Get, 345
Gear.com, 97
Google Docs, 237

H

Hojas de estilos, 283
HTML, 252
HTTP, 25
HTTPS, 25

I

If, 328
Include, 339
Include_once(), 341
Isset, 348

J

Javascript, 302, 308

M

MySQL, 358

N

Notepad ++, 314

O

ODBC, 369

P

PHP, 317
PHPMyAdmin, 59
Post, 345

R

Require, 341
Require_once(), 341

S

Sesiones, 349
SMTP, 26
Switch, 334

T

Tablas, 361
 Thinkfree, 235
 Tres niveles, 23

V

Validación, 303
 Vistas, 363

W

While, 330

X

XAMPP, 63

Z

Zoho, 233

La presente obra está dirigida a los estudiantes del Ciclo Formativo **Administración de Sistemas Informáticos en Red** de Grado Superior, en concreto para el módulo profesional **Implantación de Aplicaciones Web**.

Los contenidos incluidos en este libro abarcan los conceptos básicos sobre instalación de servidores web, explotación de gestores de contenidos, terminando con una introducción a la programación web. Se indica cómo llevarlo todo a cabo de la manera más práctica posible.

Los capítulos incluyen actividades y ejemplos con el propósito de facilitar la asimilación de los conocimientos tratados.

Así mismo, se incorporan test de conocimientos y ejercicios propuestos con la finalidad de comprobar que los objetivos de cada capítulo se han asimilado correctamente.

Además, reúne los recursos necesarios para incrementar la didáctica del libro, tales como un glosario con los términos informáticos necesarios, bibliografía y documentos para ampliación de los conocimientos.



En la página web de **Ra-Ma** (www.ra-ma.es) se encuentra disponible el material de apoyo y complementario.

